PROJECTS ■ THEORY ■ APPLICATIONS ■ CIRCUITS ■ TECHNOLOGY

NUTS AND VOLTS

EVERYTHING FOR ELECTRONICS

www.nutsvolts.com
OCTOBER 2008

Peanut Butter Monster Detector

Build this robotic bedside companion to help ease your child's night time fears

Build It:

- **PICAXE IR Multi-Board** A flexible learning tool for infrared explorations
- **Temperature Gauge** Put the heat on with temperature transducers
- **Microprocessor Controlled Wood Stove** Keep your fire burning without having to manually check the wood supply


Serial-to-Ethernet Converter
Available Now!
New 2008 Lower Pricing
PoE SitePlayer Telnet System - $99.95
SitePlayer Telnet System - $79.95
SitePlayer Telnet Modules - $24.95
SitePlayer™ TELNET
SitePlayer Telnet Module
Sample Browser Screenshot
RS232C DTE
PoE Power Over Ethernet
• Configure and upgrade through web browser
• Access micro-controller or other RS232 serial device over The Internet
• Full RS232 handshaking
• 50 to 11500 baud data rates
• DHCP, Link Local, and static IP Addresses
• TCP/IP and UDP data transfers
siteplayer.com
NetMedia Inc. Tucson AZ 85737 t.520.544.4567 f.520.544.0800 netmedia.com

Windows CE Touch Controller

CUWIN™

The CUWIN combines a graphic display and touch interface with a high efficiency industrial controller.

- 7" / 10.2" wide color TFT display
- 800 x 480 resolution, 260K colors
- Touch panel
- SD card & Ethernet support
- RS232 x 2 / RS485 x 1 or RS232 x 3
- Mono Speaker and Stereo Jack
- Real time clock (Battery backup)
- Visual Basic and Visual C++ support
- USB I/F (ActiveSync)
- Keyboard or Mouse support
- ARM9 32bit 266MHz processor
- 64MB FLASH, 64MB SDRAM
- Windows CE 5.0

* CUWIN3200 : 7" Bezel type case
* CUWIN3500 : 7" Cover Case [Front waterproof]
* CUWIN4300K : 10.2" Black Bezel type case
* CUWIN4300S : 10.2" Silver Bezel type case

CUWIN4300 $499 / Qty.1

CUWIN3200 $399 / Qty.1

COMFILE TECHNOLOGY

550 Pilgrim Drive Suite K Foster City, CA 94404
Call : 1-888-9CUBLOC (toll free)
1-888-928-2562
* 30-Day Money-Back Guarantee

Visit www.cubloc.com to see more products!

New CUBLOC Core Modules Available !

CUBLOC™

- Use BASIC and Ladder Logic
- 80 ~ 200KB Flash Data Program memory
- 2 ~ 110KB Data memory
- BASIC Speed : 36,000 instructions/sec
- RS232, SPI, I2C, MODBUS

CB320 $39 / Qty.1

$46 / Qty.1 CB380

PLC on chip with BASIC

The CUBLOC's unique multi-tasking RTOS runs BASIC and Ladder Logic programs side-by-side, allowing you to combine the flexibility of BASIC with the industry-proven power of Ladder Logic.

[Module Comparison Chart]

MODEL	CB220	CB280	CB290	[illegible]	[illegible]	CB405
FLASH	80KB	80KB	80KB	200KB	200KB	200KB
I/O	16+6	49	91	16+6	49	64
DATA	3K	3K	28K	7K	7K	110K
RS232	2	2	2	2	2	4
PWM	3	6	6	3	6	12
ETC	-	-	RTC	-	-	-

NetBurner Customer Feedback: Why are you using NetBurner?

The excellent out-of-the-box experience

I would like to comment on the quality of a product that I purchased - the MOD5270. I've purchased products in the past from other companies that claim "plug and play" functionality. Usually this means plug it in and try to figure out why it doesn't work. With the MOD5270 I literally plugged in the power, connected it to my router, and ran the factory demo.

I am also impressed with the number of example applications that are included and the fact that all the development tools are included. It was nice to be able to create a new project, import one of the examples, and run it within 5 minutes.

Keith Gilman, Sr. Design Engineer

The complete hardware and software solution

We did a previous project using different companies for the RTOS, network stack, and compiler. Yeah, three vendors and all the associated finger pointing. We had to buy the flash file system as an extra/add-on. We put an 8M flash device on our processor bus, and a guy had to write a driver for the device (data register, command register, poll the ready bit, blah, blah), it was a 3+ month job. I thought I was in for the same thing. Thanks to you, we will drop the soldered-on Flash and put on the SD connector which will be ready to roll with your file system that runs 'right out of the box'. I was sweating about all the Flash work, and now its gone.

John Ramsy, Project Engineer

The active forum filled with NB Gurus

NetBurner offers a highly versatile and affordable solution set for embedded design, remote control, and product development. From the very active user forum (with Netburner gurus in attendance), to the great IDE and tool chain. NetBurner is the first and only solution for TCP/IP-Internet-Web product automation in my book.

Chris Ruff, Software Engineer

The quality of the technical support

I've been really impressed with the quality of the tech support. The guys answering the questions are right on and super helpful (probably has to do with your policy of having the engineers handle the tickets).

Nicolae P. Costescu, Project Engineer

The full-featured tool set, examples, and documentation

The device and the NetBurner tool-set far exceeded our expectations. I spent a week familiarizing myself with the NetBurner tools, and then I wrote a first-cut of the original application that I had envisioned for the SB72EX. As it turned out, two of the example applications formed the core of the networking capabilities of my application. The bulk of my time was spent implementing a simple command interface to cause the device to interrupt its default IP-to-serial modes to receive and store a stream of data and then FTP that data to a remote server.

The NNDK interface to the IP/socket facilities of the SB72EX proved to be the simplest part of the application - the interface is straightforward and the documentation is excellent (the large collection of application examples didn't hurt). Midway through my efforts I had a couple of questions that I elected to send to the NetBurner support group - the response was almost immediate, the person(s) responding were obviously knowledgeable and experienced - how refreshing!

The SB72EX device and (perhaps more importantly) the NetBurner tool-set have proven to exceed our immediate needs and have started the gears turning as to how we might use them to address other requirements.

Fred Craft, Software Engineer

NUTS AND VOLTS

OCTOBER 2008

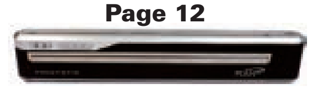
www.nutsvolts.com

Features

Projects

Columns

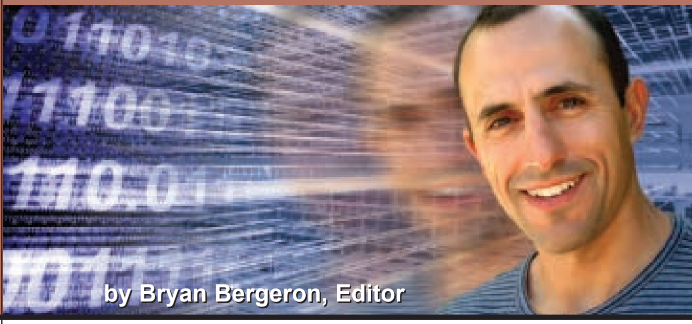
Page 12



Departments

Nuts & Volts (ISSN 1528-9885/CDN Pub Agree #40702530) is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL MAILING OFFICES. POSTMASTER: Send address changes to **Nuts & Volts, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@nutsvolts.com.


Let your geek shine.
Meet Pete Lewis, lead vocalist for the band Storytyme. Pete recently created the RS1000, a new personal monitor system for performing musicians. It was SparkFun's tutorials, products and PCB service that enabled him to take his idea to market in less than a year.
The tools are out there. Find the resources you need to let your geek shine too.
sparkfun ELECTRONICS
Sharing Ingenuity
WWW.SPARKFUN.COM
©2008 SparkFun Electronics, Inc. All rights reserved.
Hear music from Storytyme at www.storytymeband.com, or check out Pete's RS1000 at www.rockonaudio.com.



DEVELOPING PERSPECTIVES

The Man-Machine Interface — Getting It Right

If you're into electric guitars or even Guitar Hero, you're probably familiar with the iconic models from Fender and Gibson. However, there are dozens of other electronic guitar manufacturers producing everything from inexpensive, conservative clones to expensive designs on the technological edge. I recently took the plunge and purchased one of these 'edge' models: a fly mojo single cutaway made by Parker — a US electric guitar manufacturer known for its innovative designs (**www.parkerguitars.com**).

Technologically and visually, the guitar is stunning. The frame of the guitar is a hardened exoskeleton over a spruce body about half the weight and thickness of a Strat or Les Paul. Electronically, it has a traditiona magnetic pickup combined with a not-so-typical piezoelectric pickup. With this dual pickup design, I can create a tone that simulates an airy acoustic (using all piezoelectric pickup), a throaty Les Paul (using all magnetic pickup), or something in between. Furthermore, not only is the tone great, but it plays like butter.

After a few hours of playing my flame red mojo, it started to rattle. The culprit? A loose set screw in the aluminum knob on the magnetic pick-up potentiometer. I loosened the set screw and examined the potentiometer shaft. I expected a smooth, solid shaft, but what I found was a deeply slotted knurled shaft. No wonder the knob had worked its way lose.

My first thought was that this must have been a fluke or accident at the factory. After all, we're talking about a $3,200 guitar. I imagined someone in the parts department at Parker running out of solid shaft pots and running over to RadioShack to buy out their supply of potentiometers — any potentiometers that were of the appropriate resistance. However, I checked the two other potentiometers on the guitar. I didn't have to remove the knobs to see if they were of the same slotted design — the wobble of the knobs as I turned them told me the knobs and

potentiometer shafts were mismatched. In my book, the only thing worse that using smooth knobs with set screws on knurled potentiometer shafts is using a knob with a set screw on a shaft — solid or knurled — that's too small in diameter. It's virtually impossible to align a smooth knob on a knurled shaft without the knob tilting off-center. The smaller the shaft relative to the knob's interior, the worse the knob's wobble.

I briefly considered keeping the metal knobs and replacing the potentiometers with solid shaft versions from Digi-Key, but didn't want to void my warranty. Instead, I replaced the aluminum knobs with a set of Gibson push-on plastic knobs. They fit snugly and lightened the guitar by an ounce or two.

The point to the story isn't that you should avoid a Parker, but that you should pay attention to the man-machine interface in your project designs. In the case of a potentiometer, for example, it isn't enough to get the resistance, wattage, linearity, rotational life, and element composition right. You also have to think about where to best place the potentiometer on a front panel, and what kind of knob to use. It may sound trivial, but a poor quality interface leads the user to suspect poor workmanship inside. When I discovered the mismatched potentiometers and knobs, I felt as though I had purchased a little red sports car, only to have the door handle fall off after a few miles. I began to wonder what other short-cuts had been taken with the wiring inside the guitar body cavity, away from prying eyes.

Designing a good man-machine (user) interface is part art, part science, and part common sense. You might be surprised at how many electronically ingenious projects submitted for publication in *Nuts & Volts* never make it to print because the author hadn't taken the time to think about user interface. Labeling a front panel with masking tape and magic marker and adding a dozen LEDs to boot suggests the underlying circuit is little more than a hack. By the way, if you routinely mount toggle switches with knurled nuts on aluminum front panels, a simple way to avoid scratching the front pane or stripping the nuts is to use a plastic GC Nutwrench. The $18 tool is available from Stewart-MacDonald, **www.stewmac.com**. Tesco **www.Tesco.com** sells a similar, all metal unit for the same price, but I haven't worked with it. Consider these and similar tools when you're putting the finishing touch on your next creation. **NV**

NUTS AND VOLTS
EVERYTHING FOR ELECTRONICS

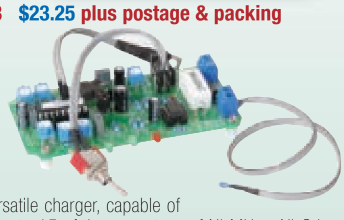
Published Monthly By
T & L Publications, Inc.
430 Princeland Ct.
Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore orders only **1-800-783-4624**
www.nutsvolts.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277
North Hollywood, CA 91615

IDEAL KITS FOR ELECTRONIC ENTHUSIASTS

FAST NI-MH BATTERY CHARGER KIT

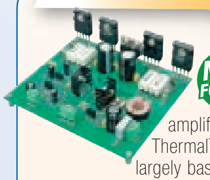
KC-5453 $23.25 plus postage & packing



A truly versatile charger, capable of handling up to 15 of the same type of Ni-MH or Ni-Cd cells. Build it to suit any size cells or cell capacity and set your own fast or trickle charge rate. It also has overcharge protection including temperature sensing. Ideal for R/C enthusiasts who burn through a lot of batteries.

- Kit includes PCB and all specified electronic components.
- Case, heatsink and battery holder not included.

AMPLIFIER MODULES



NEW FOR 08

KC-5470 $52.25 plus P&P

This ultra low distortion amplifier module uses the new ThermalTrak power transistors and is largely based on the high-performance Class-A amplifier. This improved circuit has no need for a quiescent current adjustment or a Vbe multiplier transistor and has an exceptionally low distortion figure.

- Kit supplied with PCB and all electronic components.
- Heatsink and power supply not included.
- Output Power: 135WRMS into 8 ohms and 200WRMS into 4 ohms
- Frequency Response at 1W: 4Hz to 50kHz
- Harmonic Distortion: <0.008% from 20Hz to 20kHz

KC-5471 $32.00 plus post & packing

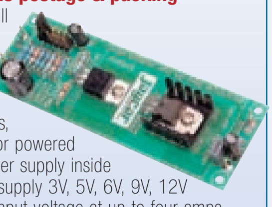
NEW FOR 08

The amp module (KC-5470) is powered using an unregulated rail only. This power supply kit is specifically designed to provide a balanced +/- 55VDC supply to power this fantastic amp kit. It has two LEDs, which illuminate when power is present on the rails and the assist in slowly discharging the filter caps when power is switched off.

- Kit includes PCB and all electronic components.

IMPROVED LOW VOLTAGE ADAPTOR

KC-5463 $8.75 plus postage & packing



This handy regulator will let you run a variety of devices such as CD, DVD or MP3 players from your car cigarette lighter sockets, even a digital camera or powered speakers from the power supply inside your PC. This unit can supply 3V, 5V, 6V, 9V, 12V or 15V from a higher input voltage at up to four amps (with suitable heatsink).

- Kit includes screen printed PCB and all specified components. Heatsink not included.

POWERTOOL BATTERY CHARGER KIT

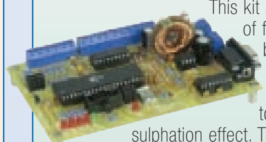
KC-5436 $23.25 plus postage & packing

Enhance the performance of the charger supplied with your power tool with this controller. It incorporates charge timeout, min and max temperature monitoring, Delta charge detection, power and charge LED indicator, adjustable Delta V, temperature settings, and optional adjustable trickle charge. Suits both Ni-Cd and Ni-MH cells.

- Kit includes PCB with overlay, case, all electronic components and clear English instructions.

BATTERY ZAPPER KIT MK II

KC-5427 $58.00 plus postage & packing



This kit attacks a common cause of failure in wet lead acid cell batteries: sulphation. The circuit produces short bursts of high level energy to reverse the damaging sulphation effect. This improved unit features a battery health checker with LED indicator, new circuit protection against badly sulphated batteries, test points for a DMM & connection for a battery charger.

- Kit includes case with screen printed lid, PCB with overlay and all electronic components with clear English instructions.
- Suitable for 6, 12 and 24V batteries
- Powered by the battery itself

TOOLS & ACCESSORIES

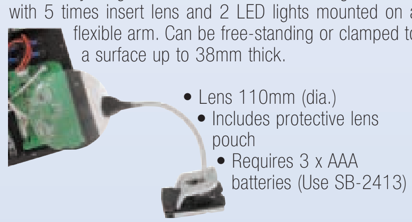
QM-1323 DMM
$17.50 plus postage & packing

A budget priced meter with everything you need - capacitance, temperature and 10A on AC and DC, compact and light weight with rugged moulded case.

- Data hold
- Temp range: -20° to 760°C
- Case included
- Category: Cat III 600V
- Display: 4000 count
- Ave/RMS: True RMS
- 137(H) x 65(W) x 35(D)mm
- For full specs see our website

QM-3532 HOBBYIST'S MAGNIFIER
$15.00 plus postage & packing

This handy magnifier has a 2 times main magnifier lens with 5 times insert lens and 2 LED lights mounted on a flexible arm. Can be free-standing or clamped to a surface up to 38mm thick.



- Lens 110mm (dia.)
- Includes protective lens pouch
- Requires 3 x AAA batteries (Use SB-2413)

TS-1112 GAS SOLDERING IRON
$15.00 plus postage & packing

Everything you need to solder; silver solder, braze, heatshrink, strip paint etc. Refillable with stand, extra tips, torch and cutting attachment.

- 3 soldering tips
- 1 torch attachment
- 1 rope cutter

FREE CATALOG

Checkout Jaycar's extensive range

We have kits & electronic projects for use in:

- Audio & Video
- Car & Automotive
- Computer
- Lighting
- Power
- Test & Meters
- Learning & Educational
- General Electronics Projects
- Gifts, Gadgets & Just for fun!

For your FREE catalog logon to www.jaycar.com/catalog or check out the range at www.jaycar.com

POST & PACKING CHARGES

Order Value	Cost
$25 - $49.99	$7.50
$50 - $99.99	$20
$100 - $199.99	$40
$200 - $499.99	$60
$500+	$740

Max weight 12lb (5kg). Heavier parcels POA. Minimum order $25.

Note: Products are despatched from Australia, so local customs duty & taxes may apply. Prices valid till 31/10/08

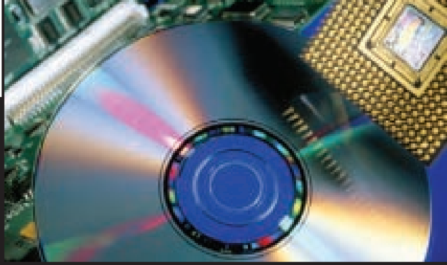
HOW TO ORDER

- ORDER ON-LINE: www.jaycar.com
- PHONE: 1-800-784-0263
- FAX: +61 2 8832 3118*
- EMAIL: techstore@jaycar.com
- POST: P.O. Box 107, Rydalmere NSW 2116 Australia
- ALL PRICING IN US DOLLARS
- MINIMUM ORDER ONLY $25

*Australian Eastern Standard Time (Monday - Friday 09.00 to 17.30 GMT + 10 hours only)
Expect 10-14 days for air parcel delivery

1-800-784-0263 www.jaycar.com

Jaycar Electronics



TECH KNOWLEDGEY 2008

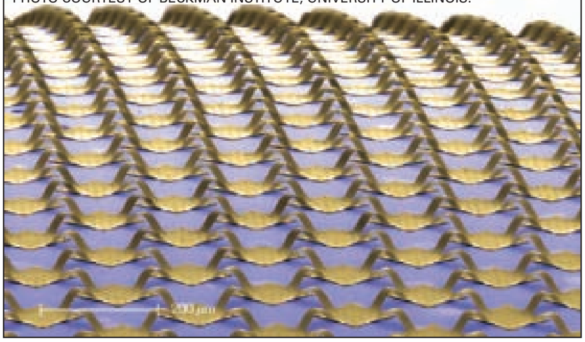
EVENTS, ADVANCES, AND NEWS

■ BY JEFF ECKERT

ADVANCED TECHNOLOGY

FLEXIBLE SENSOR WEB MIMICS EYEBALLS

PHOTO COURTESY OF BECKMAN INSTITUTE, UNIVERSITY OF ILLINOIS.



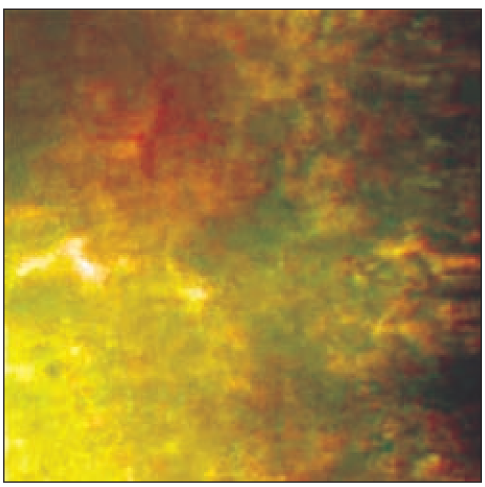
■ Low-magnification scanning electron micrograph of silicon photodetector pixels and electronics interconnected by arc-shaped ribbons.

Back in August, researchers from the University of Illinois (**www.uillinois.edu**) and Northwestern University (**www.northwestern.edu**) announced the development of a camera that works much more like the human eye than anything in current use. The secret is the sensor array, which is hemispherical in shape rather than flat. This approach eliminates some of the aberrations inherent in existing camera designs and improves the quality of captured images. The novel design replaces the traditional flat substrate with a temporary flexible backing, upon which they place an array of pixels that are interconnected by small wires.

The array, thus, can be formed into a curved shape and affixed to its permanent location on a glass lens. The array presently incorporates only 256 pixels, but the technology is based on established materials and manufacturing processes, so it should not be difficult to develop more sophisticated, higher-density devices.

Interestingly, the concept is not limited to imaging systems. Researchers are testing the same design in other applications, including a thin, conformable monitor to detect electrical signals traveling across the surface of the human brain. According to U.I.'s Prof. John Rogers, one of the project leaders, "We believe that some of the most compelling areas of future application involve the intimate, conformal integration of electronics with the human body, in ways that are inconceivable using established technologies. We are working actively with collaborators to explore possibilities in advanced health monitors, prosthetic devices, and therapeutic systems."

MAGNETIC NANOPARTICLES TO FIGHT CANCER



■ Magnetic nanoparticles capture ovarian cancer cells.

A potentially powerful cancer treatment is emerging from the School of Biology at Georgia Tech (**www.gatech.edu**). The approach is based on attaching magnetic nanoparticles to non-stationary cancer cells, capturing them, and removing them from the body. According to Gatech's John McDonald, "This technology may be of special importance in the treatment of ovarian cancer, where the malignancy is typically spread by free-floating cancer cells released from the primary tumor into the abdominal cavity." The nanoparticles are modified using a small peptide that allows them to target the tumor cells.

The technique has already been tested on mice. After staining the nanoparticles red and giving the cancer cells a fluorescent green tag, the researchers were able to simply apply a magnet and move the green cells into the abdominal region for disposal. An extra benefit is that the more accurately you target the malignant cells, the less likely you are to generate an undesirable immune response in the patient, so the treatment is more promising than using antibodies alone. The next step is to apply the concept to survival studies. The research team is also collaborating with other Gatech groups to see how peptide-directed gold nanoparticles and nanohydrogels might also be used in fighting cancer.

COMPUTERS AND NETWORKING

POCKET-SIZE PRINTER

With the ubiquity of laptops and Blackberries, it's pretty easy to take your work on the road and keep in touch with customers and associates. But what if you need a hard copy of a contract, map, boarding pass, brochure, or other document and there isn't a Kinko's in sight? The solution might be the PRINTSTIK PS910 from Planon Systems Solutions, Inc. (**www.planon.com**).

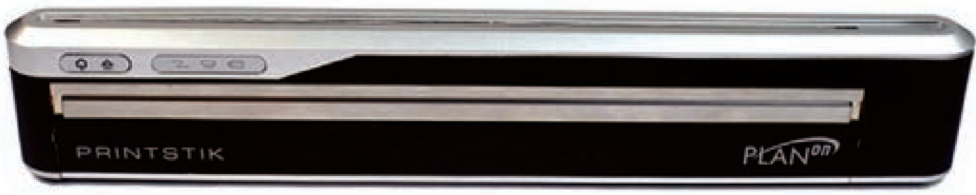


PHOTO COURTESY OF PLANON.

■ The Printstik pocket printer for PCs and Blackberries.

The PS910 is a pocket-size (1 x 10.75 x 1.5 in) thermal printer that weighs only 1 lb (460 g) and runs off its internal lithium polymer battery (or AC adapter, if an outlet is available).

You conveniently send the image to it via your PC's USB port or wirelessly using Bluetooth connectivity. Performance isn't quite up to office standards, though, with a maximum resolution of 200 x 400 dpi and throughput of "up to" 3 ppm. It also takes three hours to charge it up.

You only get 30 pages per charge, so don't expect to print *War and Peace* while sitting in your airline seat. Even less appealing is the cost, which is $299 for the printer and $24.99 for a three-pack of paper. With 20 pages per roll, you're looking at almost $0.42 per page. But if you need it, well, you need it.

STORE TERABYTES FASTER

If you're a video editor, prolific photographer, or anyone else who needs to back up and store large chunks of data, you may be interested in the second-generation Drobo product recently introduced by Data Robotics (**www.drobo.com**). Billed as the fastest box in its class for managing and storing digital information, it now incorporates FireWire 800 and "dramatically improved" USB 2.0 performance, along with redundant data protection and hot expansion up to 16 TB.

PHOTO COURTESY OF DATA ROBOTICS.

■ The Drobo automated data storage device (open front) features FireWire 800 and USB 2.0 connectivity.

Enhancements include an upgraded core processor, two FireWire 800 (FireWire 400 compatible) ports, and newly optimized firmware. It works with Windows, Mac OS, and Linux and their native file systems. The basic (i.e., no drives included) FireWire/USB four-bay Drobo lists for $499, but you can get a USB-only version for $150 less. For $899, you get the box plus two 1 TB Western Digital drives, and $1,200 gets you four drives.

FREE AIN'T FREE

Television and radio outlets are still being plastered with the creepy little dweeb singing about **freecredit report.com**, so confused consumers apparently are still falling into this Internet black hole. It's worth noting that in 2005, this subsidiary of Experian® settled FTC charges of deceptive marketing by paying reparations to deceived consumers and forfeiting $950,000 in ill-gotten gains. But nothing much has changed. The problem is that when you request a free report, you automatically are signed up for a trial membership in the "Triple Advantage" program. If you don't cancel within seven days, your credit card will begin accruing charges of $14.95 per month. But federal law requires the three credit reporting companies to provide a genuinely free report every 12 months if you ask for it. Just go to **www.annualcreditreport.com** and keep your $14.95. That way, you'll have more to spend on replica watches and enlargement pills.

CIRCUITS AND DEVICES

ELECTRIC VEHICLES ARE COMING

■ More electric vehicles — from bikes to trucks — are entering the marketplace.

With cheap oil being a thing of the past, people are increasingly taking a look at electric vehicles. Although they cannot yet compete with internal combustion in terms of power, comfort, or range, a few are looking attractive for local transportation. On the two-wheeled level is the California-built Jackal bicycle, available from a number of dealers including **www.thunderstruck-ev.com**. It offers considerably better performance than you might expect.

Powered by a 15 hp Briggs & Straton Etek motor, it has a top speed of around 45 mph (72 kph) and a range of 20 to 25 miles (32 to 40 km) on a charge. It sports some nice features like double-walled alloy rims, a hand-crafted brazed chromoly tube frame, and a choice of slick or knobby tires. Unfortunately, it costs $3,400 for the standard model and $3,700 for the performance version.

If you've got that much money to spend on a bike, you might consider upping the ante to $12,500, which gets you a Xebra Truck from ZAP! (for zero air pollution; **www.zapworld.com**). This three-wheeler goes up to 40 mph (65 kph) and 25 miles on a charge, seats two, and carries about 1,000 lb (450 kg).

Options include leather seats, an upgraded radio/CD player, and a solar panel ($1,500) for extended range. Options do not include an air conditioner or jaws of life for prying you out of a Hummer's grille.

USB RECHARGEABLE BATTERIES

This month's "Why didn't I think of that?" device is the USBCELL, from Moixa Energy Holdings. It's nothing earth-shattering, just a nickel-metal hydride cell that can be recharged from your computer's USB port, eliminating the need for an external charger. At present, only the AA size is available, but a 9V version will be available soon, and subsequent introductions will include power cells for phones, cameras, and other devices. The downside is that a pair of them costs $17.95, but think of all the alkaline batteries that won't be going into landfills.

PHOTO COURTESY OF MOIXA ENERGY.

■ These NiMH AA cells can be recharged from a standard USB port.

PHOTO COURTESY OF VITALITY, INC.

■ The GlowCap medicine bottle reminds you when it's time to pop some pills.

TAKE YOUR MEDICINE!

Let's say you're seated at the local IHOP and the rootie-tootie fresh and fruity pancakes aren't quite fruity enough. Something snaps inside, and suddenly you're screaming like a banshee and pounding your fists on the terrified waiter. Pretty soon you're in the back of a squad car, and breakfast is ruined. Obviously, you've missed your Prozac again. But take heart, because Vitality, Inc. (**www.rxvitality.com**), has introduced GlowCaps to prevent just such incidents.

GlowCaps fit standard 20-dram vials as provided by most national drug stores. All you do is insert the batteries, set your dose time, and let the cap do

INDUSTRY AND THE PROFESSION

IF YOU CAN'T BEAT 'EM, EAT 'EM

In November of 2006, IBM (**www.ibm.com**) launched a lawsuit against Platform Solutions, Inc. (**www.platformsolutions.com**), a competing mainframe vendor, accusing it of patent infringement. In 2007, PSI filed a countersuit charging IBM with antitrust violations and unfair competition. But in July, IBM tried to put an end to the unpleasantness by simply acquiring PSI for an undisclosed sum of cash. Both companies expressed delight over the merger, but not everyone was happy. Computer and Communications Industry Association (CCIA, **www.ccianet.org**) President and CEO Ed Black was quoted as saying, "This is a black-hole acquisition. It sucks the life out of the market and destroys the matter." He also predicted that this will help to perpetuate a marketplace "with little prospects for anything but complete domination by IBM." Some people are just never satisfied.

CORPORATE GAINS FOR APPLE

In 2004, Apple's share of the home computer market was down to 3.2 percent, but as of Q2 2008, it had risen to 8.5 percent. This has been a continuing trend and is not particularly surprising, but a stat that generally has gone unnoticed is that the company has a 66 percent share of machines that sell for over $1,000. Even more interesting is that the company seems to be making inroads into the corporate world in spite of marginal efforts to do so. The Yankee Group (**www.yankeegroup.com**), an independent technology research and consulting firm, recently polled 700 global IT administrators and C-level executives and discovered that 80 percent of their businesses have installed Macintosh computers. The rise is attributed mostly to the OS X operating system, but a Yankee research fellow noted, "Apple's strong marks in security, features, performance, usability, and reliability are indicative of the qualities customers value when purchasing hardware and operating system software." Maybe the droll PC vs. Mac guy ads are working after all.

the rest. Two versions are offered. The GlowCap SOLO is a self-contained unit that flashes orange and plays a melody ("Lucy in the Sky with Diamonds?") when it's time to medicate. But if you want to fly first class, try a GlowCap CONNECT™, which includes a wireless modem that connects to the Vitality network. The cap collects data on your drug use or lack thereof so Vitality can phone you if you forget to take a medication, send weekly emails to a family member, and issue monthly adherence reports to you and your caregiver. Heartily endorsed by Britney Spears, Lindsay Lohan, Charlie Sheen, et al., the caps should be appearing in pharmacies soon. **NV**

■ BY LOUIS E. FRENZEL W5LEF

THE SOFTWARE-DEFINED RADIO IS REAL

Trading Hardware for Software Makes for the Ultimate in Versatility

WE ALL THINK OF A RADIO AS A PHYSICAL DEVICE. It used to be a big wooden box with tubes in it, but the radio morphed into a small plastic box with transistors and today it is — in most cases — a single integrated circuit.

But now, many of the functions previously performed by electronic components and circuits are being carried out by special programs running on a fast digital processor or, in some cases, a fast programmable logic device. While the radio is still a form of hardware, that hardware is programmable so the radio will do what you want. This is not just some fantasy. It is just as real as the cell phone in your pocket. Here's an update on this hot wireless technology.

SDR REFRESHER

In case you forgot, or actually in case you never knew, a software-defined radio (SDR) is considered to be any radio receiver or transmitter that uses software to define some of the basic physical functions usually carried out by traditional circuitry. (Think filtering and demodulation.) And since we are saying that software defines the various characteristics of the radio, that implies the use of some form of microcomputer that executes the software. In some cases, that micro is a special digital signal processor (DSP) or at least an embedded controller fast enough to do the functions in real time. It could also be a regular PC or laptop.

Many new SDRs incorporate a programmable logic device like a field programmable gate array (FPGA). The bottom line is the radio essentially just becomes another electronic product based on a micro. What product does NOT have an embedded controller or other processor in it? For discussion purposes in this article, I am only going to cover receivers. Transmitters are a special case.

So, can I just connect an antenna directly to a microcomputer and make a receiver with the proper software? The answer is, almost. In fact, the simplest SDR receiver is almost just that. Since radio signals are very low level, they almost always must be amplified before you do any processing on them. So, a simple SDR would start with an RF amplifier after the antenna.

Don't forget that radio signals are analog signals. Microcomputers are digital and process binary information. So, after the RF amplifier you have to use an analog-to-digital converter (ADC) to translate the analog RF signals into a stream of binary numbers that the micro can process. With the digitized RF in memory, the processor can go to work performing all the remaining functions of a receiver.

The two basic receiver functions are frequency selection (or what we also call tuning) and demodulation. Both of these functions can be done in software. Frequency selection involves filtering, typically using some sort of band pass filter to distinguish one signal from another. The filter selects only the one we want and diminishes the others.

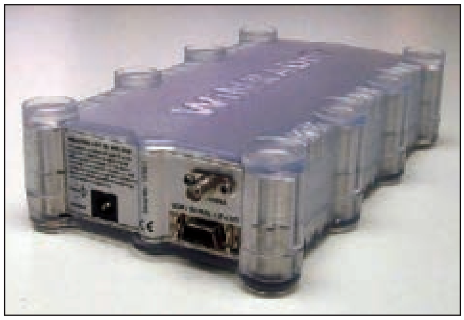
Once the signal is selected, we put it through a demodulation process to recover the originally transmitted information. If that information was voice, then after the demodulation process we have a file of binary data representing that voice. Now all we do is send that data to a digital-to-analog converter (DAC) that gives us the voice in natural analog form. Usually, the DAC output must be amplified by an external single chip audio amplifier capable of driving a speaker or a headset. The big issue in all of this is how fast we can perform analog-to-digital conversion. To accurately represent the analog data, we must sample it at a rate at least twice the highest frequency content. If we want to receive signal up to say 30 MHz, then our ADC needs to run at least at 60 megasamples per second

(MS/s) and preferably faster. ADCs capable of that speed are readily available.

If you want to receive higher frequencies, you will need a faster ADC. You can buy ADCs with a sampling rate up to several hundred MS/s, but they are expensive. In fact, some ADCs can sample at a gigasample per second (GS/s). Digital oscilloscopes and other digital test instruments use the fast ADCs but as I mentioned, they are expensive for receivers. The technique used to eliminate the need for a super fast ADC is to downconvert the input RF signal to a lower intermediate frequency (IF). That is the superheterodyne process used in most modern receivers of the immediate past. For example, many FM radios convert an FM station on 98.7 MHz down to 10.7 MHz IF. That is easily digitized by a modestly priced ADC. This process works great but requires extra circuitry such as a mixer after the RF amplifier and a local oscillator for tuning and frequency selection. Another technique is to downconvert the signal directly to the frequency range of the modulation signal. That is called the baseband signal. By using a mixer and a local oscillator tuned to the signal frequency, the output will be the original data. We can then take out the RF with a simple low pass filter and send the rest to the ADC. This technique is referred to as direct conversion.

Direct conversion only works with amplitude modulation so to make it applicable to other forms of modulation, we use a pair of mixers which each get the RF signal from the amplifier. Each mixer is then driven by the local oscillator, but with a 90 degree phase shift between the two. This generates what we call the in-phase (I) and quadrature (Q) signals at the mixer outputs. The I and Q signals are each sent to an ADC. Each ADC creates a stream of binary data representing the original baseband signal like voice or video.

To demodulate anything other than AM such as FM, FSK, PSK, or QAM (or any other more exotic form of modulation), both the I and Q signals are needed. These two signals contain the amplitude and phase information needed by the software algorithms that are used to demodulate the signals. And incidentally, virtually all cell phones use this approach and most other wireless devices have also adopted it. So, all are what we can legitimately call SDRs. The final issue of concern is whether our processor that does the filtering and demodulation can keep up with the data and perform in real time. High speed is certainly necessary; special DSPs are made for SDR with a clock speed of 1 GHz or more. And, of course, you can also use a PC to do these functions. The modern PC runs plenty fast for most basic SDR operations.



■ FIGURE 1. The WiNRadio doesn't even look like a radio, but is a full superhet with DSP filtering and demodulation. It covers the 9 kHz to 30 MHz range with all modes.

PRACTICAL SDRS

I have already said that virtually all cell phones are SDR. The newer TV sets — especially HDTVs — are SDR. The new HD radios are SDR. And the military is a big user of SDRs. More and more radios are becoming SDR in format such as amateur radio equipment and shortwave radios.

The military is especially enamored with SDR because it gives a radio a ton of flexibility. One set of hardware can be used to make a radio with different frequencies of operation, as well as different modulation schemes. To make a change, all you do is select the appropriate piece of software stored in memory and away you go. You can change on-the-fly, in fact. SDRs are amazingly versatile as they are really many radios in one.

Can you buy an SDR? You bet! And chances are you already have one (like your cell phone). Commercial radios especially for amateur radio use or shortwave listening are available. A good example of a commercial SDR for SW is the WiNRadio WR-G303e shown in Figure 1. It covers the range of 9 kHz to 30 MHz. It uses standard superheterodyne techniques with IFs of 45 MHz and 12 kHz. After the ADC, DSP filters take care of the selectivity and demodulation for AM, SSB, CW (Morse code), and FM.

What you are probably wondering is where is the front panel with all the knobs, dials, and switches? Answer: There is none. The front panel is also implemented in software. That is done on an external PC or laptop plugged into the USB port on the radio. Figure 2 shows a graphical user interface front panel where the mouse and the keyboard control the frequency selection, function, volume, etc. Note that the panel includes a spectrum analyzer function so you can see a frequency display of the signals being received. A fast Fourier transform (FFT) generates that frequency domain display. The FFT is really a mathematical algorithm that takes samples from a time domain signal (amplitude variations over time) and translates it into a display of signal amplitude vs. frequency. Another SW receiver using SDR is made by

■ FIGURE 2. This is the virtual front panel of the WiNRadio displayed on a PC to which the radio is connected.

■ FIGURE 3.The Flex-Radio FLEX-5000A is the ultimate SDR ham radio for bands from 160 to six meters. The DSP is performed on a PC. The front panel is implemented on the PC with the PowerSDR software. The transmitter output is 100 watts.

Perseus. It does not use the superhet or direct conversion methods just described, but directly digitizes the input signal. Some standard inductor-capacitor (LC) filters after the antenna select the frequency band of interest. The signals in that band are then sent to an RF amplifier and then to a 14-bit 80 MS/s ADC. Next, the digitized signals go to a fast FPGA where further filtering is performed, as well as demodulation. The output then goes to a PC for display with a front panel similar to that shown in Figure 2.

If you want to play around with an SDR receiver, there are several kits and packages available. One example is the QuickSilver receiver. It uses direct conversion to get the baseband signal followed by a fast ADC. It covers the 15 kHz to 55 MHz range. An Altera FPGA does the DSP for filtering and demodulation. Another interesting SDR receiver is called the SoftRock. It is complete on a small PC board attached to a USB connector. The receiver front-end is on that board which plugs into a PC. The receiver produces the I and Q signals that then go to the PC sound card or audio inputs. The sound card ADC conversion is used to generate the digital data. Software that runs on the PC itself does the DSP for filtering and demodulation.

A small company called RF Space makes the SDR-IQ radio. It covers the 500 kHz to 30 MHz range. Both use an extensive software package called SpectraVue that runs on a PC to do all the SDR functions and to implement the front panel complete with frequency spectrum displays.

FOR MORE INFO ON SDR

FlexRadio Systems
www.flex-radio.com

GNU Radio
www.gnu.com

High Performance SDR
www.hpsdr.org

PERSEUS
www.ssb.de

QuickSilver
www.philcovington.com/QuickSilver

RF Space
www.rfspace.com

SoftRock
www.softrockradio.org

Software-Defined Radio Forum
www.sdrforum.org

Tucson Amateur Packet Radio Corp.
www.tapr.org

WiNRadio Communications
www.winradio.com

AMATEUR RADIO SDR

Amateur radio has generally been slow to adopt SDR but most manufacturers have radios now with more advanced SDR products in the works. Most ham transceivers at least incorporate DSP for filtering and demodulation. But there is one ompany that has gone all the way with a SDR. That is Flex Radio. Their FLEX-5000A is a full-blown SDR transceiver (see Figure 3). This amazing radio covers all ham bands from 160 meters to six meters in one box and you can buy a transverter that adds coverage of two meters, as well as 70 cm (440 MHz). It will operate in any mode — CW, AM, SSB, FM, and even RTTY (teletype) and the popular digital mode PSK31. The transmitter puts out 100 watts of power.

The receiver uses direct conversion and develops the I and Q signals that go to internal ADCs and then on to a connected PC. A fast PC does all the DSP and other signal processing. That software is called PowerSDR and handles all receiver and transmitter signal processing. It also provides an impressive front panel display that features the frequency spectrum display, as well as the impressive waterfall display that shows frequency horizontally, time vertically, and signal strength in different colors. The FLEX-5000A is impressive, but expensive. It shows that SDR is practical — it delivers a transceiver whose specs are difficult to duplicate with older technology. A more modest effort in ham SDR is the work of the High Performance SDR (HPSDR) project. This is an informal group of engineers, techs, programmers, and others who work together to design and build modular SDR items. The products are on a small PC board that plugs into a back plane like that shown in Figure 4. One board is an ADC, another is a PLD or FPGA for the DSP, and others implement the transmit functions. There are a whole series of these boards you can buy to build an SDR receiver. Most are set up to use Flex Radio's PowerSDR software which is an open source package available for free on the Flex Radio website; check it out for more details on the HPSDR modules.

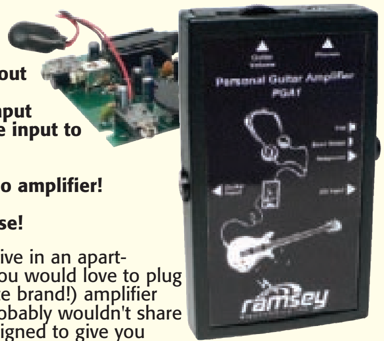
SDR is still complex and a bit expensive, but it is the future of radio. The days of the crystal set, one tube radio, or transistor radio kit are over. It is time to learn more about SDR with one of these products. **NV**

■ FIGURE 4. High Performance SDR products modularize all segments of an SDR receiver and transmitter allowing you to experiment, as well as mix and match.

The Latest & Greatest!

The Newest and Neatest Stuff!

Personal Practice Guitar Amplifier



- ✔ Practice your guitar or bass without disturbing others!
- ✔ Standard ¼" guitar audio plug input
- ✔ CD/MP3 player audio headphone input to practice to your favorite music!
- ✔ Switchable bass boost!
- ✔ Great as a DI to your home stereo amplifier!
- ✔ Uses standard 9V battery
- ✔ Includes guitar strap/belt clip case!

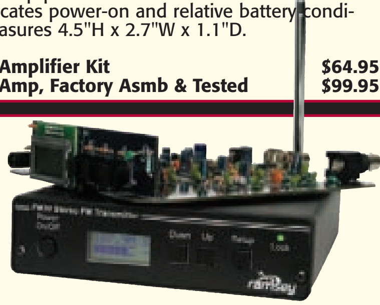
So you're a guitar musician and you live in an apartment...what else needs to be said! You would love to plug into your 100 watt HiWatt (my favorite brand!) amplifier and let it crank, but the neighbors probably wouldn't share your enthusiasm! The PGA1 was designed to give you hours of enjoyment playing your favorite electric guitar while isolating your family, friends, and neighbors from your music and your passion. *(I still use the original engineering prototype with my SG with P90's!)*

Whether you use an acoustic electric, electric, or bass guitar, the PGA1 gives you an adjustable volume output to your favorite headphones. Unlike a lot of practice amps, the PGA1 has a special bass boost to enhance bass guitar use. One practice amp for all your guitars! If you want to practice to your favorite music, the PGA1 has a standard style audio input to play to your favorite music! Just plug in your Walkman style CD player, MP3 player, or even the headphone output on your mixer board and you'll hear both your music source and your guitar in your headphones!

The PGA1 uses a powerful LM386 power amplifier to provide one watt of power to your headphones or even a small speaker! It gets even better...use the PGA1 as a cheap and easy direct inserter box to match your guitar into an aux input of a stereo receiver amplifier or PA amplifier! It sounds GREAT! A standard ¼" audio input connector is used for your guitar input, matching your standard guitar cables. A top mounted on/off/volume control adjusts the headphone output level through the full range. The small belt clip pocket size device runs on a standard 9V battery (not included). A LED indicates power-on and relative battery condition. The attractive compact case measures 4.5"H x 2.7"W x 1.1"D.

PGA1	**Personal Practice Guitar Amplifier Kit**	**$64.95**
PGA1WT	**Personal Practice Guitar Amp, Factory Asmb & Tested**	**$99.95**

Digital Conrolled FM Stereo Transmitters



- ✔ Rock stable PLL synthesized
- ✔ Front panel digital control and display of all parameters!
- ✔ Professional metal case
- ✔ Super audio quality!
- ✔ 25mW and 1W models!

For nearly a decade we've been the leader in hobbyist FM radio transmitters. When it became time for a new model, we started from the ground up! We told our engineers we wanted a new technology transmitter that would provide FM100 series quality without the advanced mixer features. They took it as a challenge and designed not one, but TWO transmitters!

The FM30 is designed using through-hole technology and components and is available only as a do-it-yourself kit, with a 25mW output very similar to our FM25 series. Then the engineers redesigned their brand-new design using surface mount technology (SMT) for a very special factory assembled and tested FM35WT version, with 1W output for our export-only market! Both are designed around an RF tight vinyl clad metal enclosure for noise free and interference free operation. All settings are done through the front panel digital control and LCD display and they are stored in non-volatile memory for future use. Both the FM30 and FM35WT operate on 13.8 to 16VDC and include a 15VDC 110/220VAC plug in power supply. The stylish black anodized aluminum case measures 5.55"W x 6.45"D x 1.5"H. and is a great match to your other equipment.

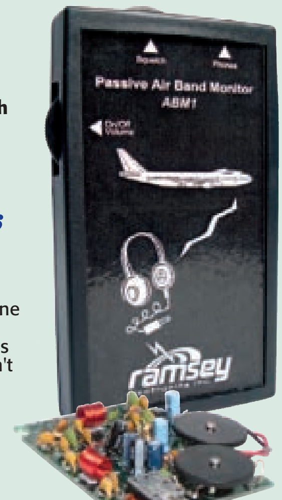
(Note: After assembly of this do-it-yourself hobby kit, the user is responsible for complying with all FCC rules & regulations within the US, or any regulations of their respective governing body. FM35BWT is for export use and can only be shipped to locations outside the continental US or valid APO/FPO addresses or valid customs brokers for end delivery outside the continental US.)

FM30B	**Digital FM Stereo Transmitter Kit, 0-25mW, Black**	**$199.95**
FM35BWT	**Digital FM Stereo Transmitter, Assembled, 1W, Black**	**$299.95**

Get The Catalog!

Get the latest 2008 Ramsey Hobby Catalog! ***112 value packed pages*** of the neatest goodies around with lots of new stuff! Order yours today on line or give us a call... Or download the PDF at www.ramseykits.com/catalog!

Passive Air Band Aircraft Monitor



- ✔ Monitors the entire aircraft band without any tuning!
- ✔ Passive design, can be used on aircraft, with no local oscillator, it generates and creates no possible interference!
- ✔ Great for air shows
- ✔ Patented circuit and design!

Written Up In Communications Magazines Worldwide!

For decades we have been known for our novel and creative product designs. Well, check this one out! An aircraft receiver that receives all nearby traffic without any tuning. It gets better... there is no local oscillator so it doesn't produce, and can't produce, any interference associated with all other receivers with an LO. That means you can use it onboard aircraft as a passive device! And what will you hear? The closest and strongest traffic, mainly, the plane you're sitting in! How unique is this? We have a patent on it, and that says it all!

This broadband radio monitors transmissions over the entire aircraft band of 118-136 MHz. The way it works is simple. Strongest man wins! The strongest signal within the pass band of the radio will be heard. And unlike the FM capture effect, multiple aircraft signals will be heard simultaneously with the strongest one the loudest! And that means the aircraft closest to you, and the towers closest to you! All without any tuning or looking up frequencies! So, where would this come in handy?

1. ***At an air show! Just imagine listening to all the traffic as it happens***
2. ***Onboard aircraft to listen to that aircraft and associated control towers***
3. ***Private pilots to monitor ATIS and other field traffic during preflight activities (saves Hobbs time!)***
4. ***Commercial pilots to monitor ATIS and other field traffic as needed at their convenience***
5. ***General aircraft monitoring enthusiasts***

Wait, you can't use a radio receiver onboard aircraft because they contain a local oscillator that could generate interfering signals! We have you covered on that one. The ABM1 has no local oscillator, it doesn't, can't, and won't generate any RF whatsoever! That's why our patent abstract is titled "Aircraft band radio receiver which does not radiate interfering signals". It doesn't get any plainer than that!

SPECIFICATIONS

Frequency Range:	118 MHz to 136 MHz
Receiver Type:	Patented Passive Detector
IF Frequencies:	None!
Receiver Sensitivity:	Less than 2 uV for detectable audio
Audio Output:	700mW, 8-24 ohms
Headphone Jack:	3.5mm stereo phone, stereo earbuds included
External Antenna:	Headphone cord coupled
Power Requirement:	9VDC battery
Dimensions:	2.25" x 2.8" PC Board 2.5" x 4.6" x .9" Case
Weight:	4 oz. with battery

Even with its compact design, we designed the ABM1 for easy construction by the electronic hobbyist. Thru-hole parts are used for all customer assembly. Several SMT components are part of the circuit, and they are preinstalled at the factory! In addition, a factory assembled and tested SMT version with additional shielding for adjacent channel overloading is also available for the aircraft professional that just wants to get listening quickly. Stereo earbuds are even included! If you're a private pilot, commercial pilot, airport worker, aircraft enthusiast, attend air shows, or spend a lot of time in airports or onboard aircraft, the ABM1 is for you!

ABM1	**Passive Air Band Monitor Kit**	**$89.95**
ABM1WT	**Passive Air Band Monitor, Factory Assembled & Tested**	**$159.95**

MORE than just friendly on-line ordering!

Clearance Specials, Ramsey Museum, User Forums, Dealer Information, FAQ's, FCC Info, Kit Building Guides, Downloads, Live Weather, Live Webcams, and much more!

www.ramseykits.com

800-446-2295

ramsey
www.ramseykits.com
590 Fishers Station Drive
Victor, NY 14564
(800) 446-2295
(585) 924-4560

Where Electronics Is Always FUN!

- ✦ ***Build It!***
- ✦ ***Learn It!***
- ✦ ***Achieve It!***
- ✦ ***Enjoy It!***

Q&A

WHAT'S UP:

Join us as we delve into the basics of electronics as applied to every day problems, like:

- ✔ Solar Power Supply for Tube Radio
- ✔ TV Converter Box
- ✔ Night Light Control

■ WITH RUSSELL KINCAID

In this column, I answer questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions.

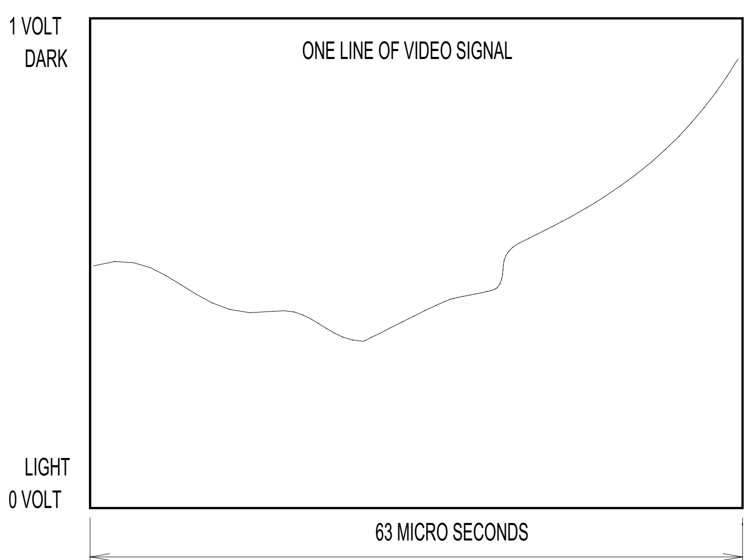
Send all questions and comments to:
Q&A@nutsvolts.com

VIDEO MOTION DETECTION

Q Is there a simple way to tell when a video camera sees motion, so that a switch can be closed to start a VCR or DVD recorder? The camera output is nominally one volt p-p.

A The short answer is no. Let me explain: The video camera scans the scene with 500+ lines. First, it scans all the odd lines in 1/60th of a second, then it scans all the even lines in 1/60th of a second, resulting in a frame rate of 30 Hz.

Figure 1 is an example of a typical scan line. The amplitude of the signal depends on the brightness of the object being scanned; white is zero volts and black is one volt (for the American market, I don't know about others). Each line is slightly different so you would have to compare line 128 of frame A with line 128 of frame B to tell if there was a difference. We are now talking about digital signal processing which is out of my field and is not simple.



■ FIGURE 1

PIC BASIC COMPILER

Q I like your idea of using the PIC12F675 as a timer. What would the code be if written in C for the 15 minute timer in the Q&A section? I use MPLAB IDE and it doesn't have a Basic compiler that I am aware of. Also, would it be possible to use this chip for a timer that is two hours or longer? When I write time-delay with for loops in C, I can't get the delay longer than a few seconds without the program hanging (i.e., I have a light flash and delay the on-off with a for loop). As I slowly increment the for loop, there becomes a point where the light doesn't flash in a delayed response like it should).

— Rich Bertelsen

A I can't help you with the code in C, but there is a Basic compiler called PICBASIC PRO. You get MicroCode Studio with it, which greatly simplifies the process. You can download a free demo version from **www.mecanique.co.uk/products/compiler/pbp-demo-index.html**. The demo version has a limit of 31 lines of code and only supports 16F84, 16F62x, and 16F87x; 12F675 is not supported.

There is essentially no limit to delay time; you can delay 65,535 milliseconds in one statement but if that is not enough, you can do it again and again. You can also put it to sleep and use the watch dog timer to wake it up again.

The problem with your for-next loop could be in the dimension of the count symbol. There are so many possibilities, your best bet is to post your code on one of the forums. The

Nuts & Volts forum will be helpful (**http://forum.servomagazine.com/index.php**).

NIGHT LIGHT CONTROL

Q I saw the article in the July *N&V* about using a PIC12F675 to power off a circuit feeding a camera and solar-powered fountain pump when night falls. It is almost exactly what I need, except I need the circuit to power up some 120 volt lights at dusk. Also, I would like to be able to turn them off at some later time (from two to five or six hours) and have the circuit ready for the next nightfall.

I suppose the code would have to be changed somewhat, and possibly the LDR, as well. I have a Picstart-Plus programmer, although it hasn't been used for several years, I think it will still work. Anyway, I certainly would appreciate some input about the subject.

— Bert Russell

A You will need a five volt supply and the circuit in Figure 2. I used one of the large LDRs from the RadioShack assortment (276-1657). You will want to adjust R2 such that the light comes on when it is sufficiently dark. A three position switch selects the light on time to be two, five, or seven hours. The opto-triac 653-G3MB-202P-DC5 (Mouser; **www.mouser.com**) is good for two amps; if you need more current, you can use 881-LS24D16C-HS1 which is rated 240 volts, 16A.

The program is listed in Figure 3. The program loops around the first statement as long as it is daylight. At night, the program jumps to two, five, or seven hours delay, depending on the SW1 position. The final delay is 13 hours to get back to daylight and start over.

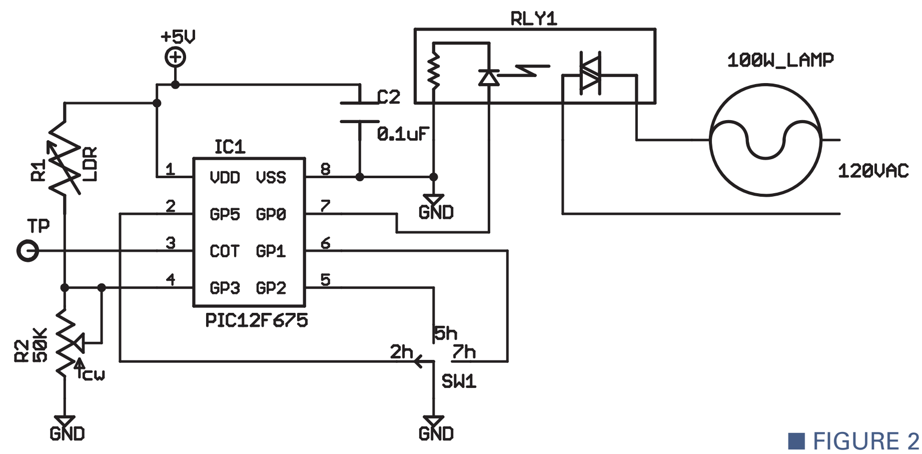


FIGURE 2

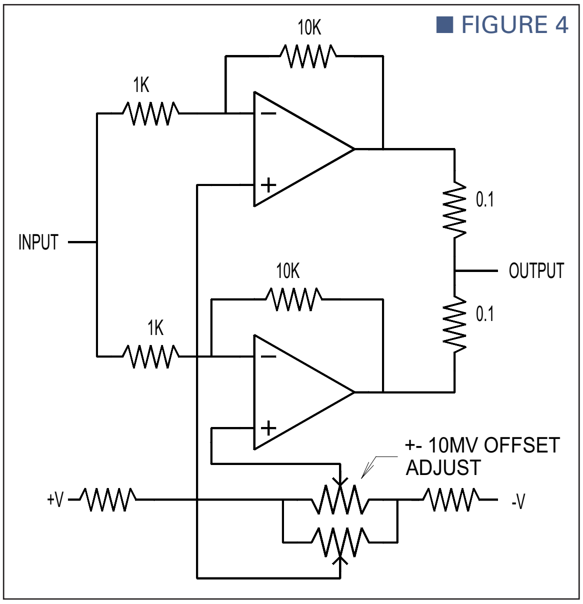
```
'***************************************************************
'*  Name    : NIGHT LIGHT CONTROL.BAS                          *
'*  Author  : Russ Kincaid                                     *
'*  Date    : 7/11/2008                                        *
'*  Version : 1.0                                              *
'*  Notes   :                                                  *
'*  The micro sleeps during the day and turns on a light when *
'*  darkness falls.  The light stays on for 2, 5, or 7 hours  *
'*  depending on a 3 position switch that grounds GP1, GP2, or *
'*  GP5.  you can use 12F629 which is the same device withour *
'*  The A/D but remove the ANSEL statement or the compiler will *
'*  balk. GPIO.4 (pin3) is oscillator output test point.      *
'***************************************************************
    REM DEVICE = 12F675
    CMCON = 7                   'SETS DIGITAL MODE
    ANSEL = 0                   'GPIO.0 TO GPIO.3 SET AS DIGITAL
    TRISIO = %00101110          'GPIO.0 AND GPIO.4 AS OUTPUT
    VRCON.7 = 0                 'TURN OFF VOLTAGE REFERENCE
    DEFINE OSCCAL_1K 1          'TO SAVE OSCILLATOR CALIBRATION
    OPTION_REG = 0              ' GLOBAL PULLUPS ENABLED
    J VAR word
 '    ***************************************************************
    START:

        IF GPIO.3 = 1 THEN START
        HIGH GPIO.0             'TURN ON TRANSISTOR SWITCH
        IF GPIO.5 = 0 THEN TWO
        IF GPIO.2 = 0 THEN FIVE
        IF GPIO.1 = 0 THEN SEVEN

    TWO:
        FOR J = 1 TO 120
        PAUSE 60000             '120 TIMES 1 MINUTE = 2 HOURS
        NEXT J
        GOTO DAY

    FIVE:
        FOR J = 1 TO 300
        PAUSE 60000
        NEXT J
        GOTO DAY

    SEVEN:
        FOR J = 1 TO 420
        PAUSE 60000
        NEXT J
        GOTO DAY

    DAY:

        LOW GPIO.0              'TURN OFF TRANSISTOR SWITCH
        for j = 1 to 780
        pause 60000             'PAUSE 13 HOURS, UNTIL NEXT DAY
        next j
        goto start
        END
```

FIGURE 3

PARALLEL OP-AMPS

Q Can I safely parallel LM675 op-amps?

— Craig Kendrick Sellen



■ FIGURE 4

A Not directly because the input offsets will be different. Two op-amps on the same chip have the same offset, so you can do it there, but these are separate chips. Figure 4 is a way to do it. You will lose a maximum of 0.3 volts of output swing, but the current capacity will be double. In the typical case, you could get away with no offset trim, but you should always design for the worst case (10 mV offset).

SERVICE MANUAL FOR TV

Q I have a Zenith (now LG) DLP Projection TV model Z44SZ80 that no longer will power on (after a power surge). Before sending it to the recycle bin, I would like to try to fix it, but LG says that their service manuals are available only to "authorized service dealers." Do you know of some service like Sam's PhotoFact of years past where I might get service info for this item?

— Len

A A service manual is available from **www.rtv-horvat-dj.hr/apl/index.htm**. It will cost $10 US.

TV CONVERTER BOX

Q I purchased a DataStream TV converter box from RadioShack. With rabbit ears, a preamp, and a VHS in series I get 10 channels across the VHF and UHF bands. I connected the converter just after the antenna and also tried just before the TV. Only one station came in and did so very well in the latter configuration. The other channels said weak signal. Eliminating the preamp and VHS player did not work either.

Questions:
1) Does digital TV require a stronger signal?
2) Do I need to update my antenna?
3) Do you have a better recommendation for installation?
3) Should I return the converter and buy another brand?
4) Or, should I scrap the converter and get cable or DirecTV?

— Drew

A 1) Digital TV does require a better signal to noise ratio. The phase lock loop in the analog TV will stay locked even when down in the noise but the digital converter can't tell the difference between noise and signal, so has to have signal well above the noise.

2 and 3) I would not expect digital TV to work from rabbit ears unless the transmitter was very close. I recommend a roof mounted antenna with amplifier and rotator. If that is not feasible, look into a wide band helical antenna that will fit inside.

4) Cable will provide more channels and options, and will be more reliable than the antenna, but costs more. The choice is up to you.

BATTERY CHARGER

Q I am a subscriber to *Nuts & Volts* and I LOVE your Q&A column. I work in a failure analysis lab; the reason I am writing is that we will soon be getting in an old battery charger (see Figure 5). The charger is designed to charge a bank of large lead-acid batteries. The batteries are in parallel and have a float voltage of about 36 volts.

I understand the purpose of T2, L2, D1, and D2 — it is a simple full wave rectifier. But what is the purpose of T1, C1, and L1? Why are they doing that? Any ideas?

— Michael Craft, Saint Paris, OH

MAILBAG

Dear Russell,

In the Q&A section of the August '08 issue, the schematic in Figure 6 and Figure 8 show a relay as the collector load of a 2N2222 transistor. As you know, when the transistor turns off, the energy stored in the inductance of the relay coil will cause a voltage spike. Since there is no snubber or diode across the coil, the only way for the inductance to discharge is by causing the collector-base junction to conduct as a zener.

Maybe there is not enough energy in the relay coil to destroy the transistor, but I would think repeatedly causing collector-base breakdown would have some negative impact on reliability.

— Jim Stewart

Dear Russell,

There should be a diode placed in parallel with RLY1 with its cathode connected closest to the positive supply and the anode placed at the other end of the relay coil. The diode should not be switched out of the circuit by any of the control wiring.

With no diode present, the back EMF of the field collapse when power is removed from the relay will cause the transistor to fail and control of the relay will be lost.

A diode such as a 1N4004 should do the trick.

— Mike Flood

Response: In my defense, the diode is not needed in Figure 6 because the solar cell turns the transistor off very slowly, so Ldl/dT will be small. In Figure 8, I added a switch that will turn the transistor off rapidly and the diode is needed. Mea culpa once again.

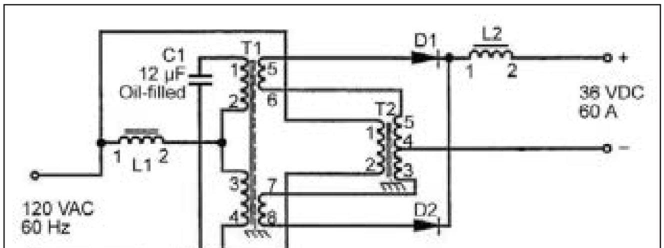
A Here is my thought: T1 is evidently a ferro-resonant regulating transformer, L1 increases the source impedance making it look more like a current source and limiting the charging current. T2 is a low impedance standard transformer that supplies the float voltage. I don't know why T2 was not a regulating transformer also. L2 is not needed because batteries don't care about being pulsed; in fact, it may be better than smooth DC.

SOLAR POWER SUPPLY FOR TUBE RADIO

Q I would like to build a power supply that will allow me to run a tube-type portable radio using a solar panel. My radio (a Motorola 56B1) requires 6.5V @ 50 mA to power its filaments and 90V @ 8.5 mA for its plate supply. (The radio's total power consumption is 1.09W.) Most solar panels available from places like Silicon Solar have open circuit voltages of 8V, so that will probably be the voltage I will have to work with. The 90V supply is going to be the hardest to create. Do you know of any circuits that could generate these voltages? Any assistance you can provide will be greatly appreciated! If possible, I would like to use parts available in through-hole mounting packages please.

(I know in your April issue you had an article on building an HV switching power supply, but I don't think it would be efficient enough for my application.)

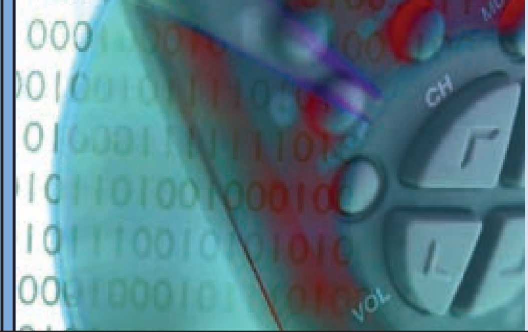
— Matthew S. Taylor



■ FIGURE 5

A I breadboarded the circuit in Figure 6; the efficiency was 68% — not very good. The solar panel supplies 90 mA in bright sun, which is not enough to power the radio directly, but two in parallel would keep the battery charged. The battery is needed so the radio does not die when a cloud comes by, and it regulates the voltage. The solar panel will recharge a dead battery in a couple of sunny days with the radio off. I did not put a diode in series with the solar panel because the solar panel is a diode. I am using a panel to trickle charge 12 volt batteries and there is no leakage at night. Some low quality solar cells may have significant leakage, so it is a good idea to check. The 556 dual timer provides an almost 50% duty cycle drive to the MOSFET, Q1, because the second stage ramp time is 1/2 the first stage. The first stage is free running at 60 kHz and triggers the second stage one shot. The feedback pulls down the CV pin of the one shot, reducing the pulse width to control the DC output. The red LED is the voltage reference. An LED is a better regulator than a low voltage zener.

The transformer uses a ferrite E core from Magnetics, Inc., part number F-42510-EC. The bobbin part number is B2510-01. The primary is 10 turns #27 wire and the secondary is 160 turns #36 wire. The calculated secondary turns would be: 10 turns * 90 volts/6 volts = 150 turns, but I added 10 turns to allow for regulation. I had planned to operate at a higher frequency, but found that the transformer resonated at 65 kHz, so I set the frequency at 60 kHz. The regulation is not great; when adjusted for 90 volts with 10K load, the no load voltage is 100 volts. Perhaps I should have added 20 turns in order to make the regulator work harder. The parts list is in Figure 7. **NV**



NEW PRODUCTS

- HARDWARE
- SOFTWARE
- GADGETS
- TOOLS

GUARDIAN AL
G9 LASER SYSTEM

A-Elektronik Company has unveiled a successor to the long line of Guardian laser-based driving aids: the Guardian AL G9.

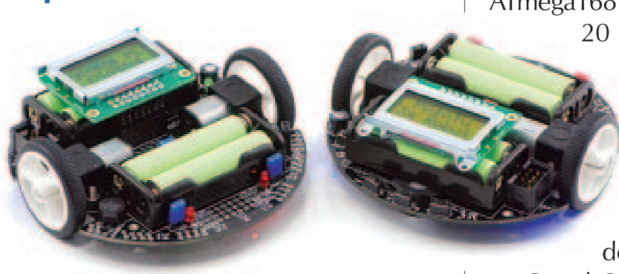
The AL G9 is a significant step forward in the industry of laser-based multifunctional driving aids. The technology is based on its very popular predecessor (the AL G8) which itself made breakthroughs in laser detection and interference blocking, and was branded the "top dog" on several tests. The Guardian AL G9 now brings improved laser detection, thanks to patent-pending detection technology.

The AL G9 sensing circuit is now microcontroller controlled. The microcontroller is monitoring the operating temperature of the device and at the same the level of Sun saturation of the receiving diodes. Together with a sophisticated algorithm and pre-stored data in the memory the microcontroller has allowed the reception sensitivity to reach near maximum.

This and other improvements have allowed the AL G9 to be smaller, smarter and more accurate driving aid, also the question of buyers budget is addressed hence where a vehicle before required two sensors per side for proper protection now only one sensor is necessary.

For more information, contact:
A-Elektronik Company
Web: **www.a-elektronik.hr**

3pi ROBOT

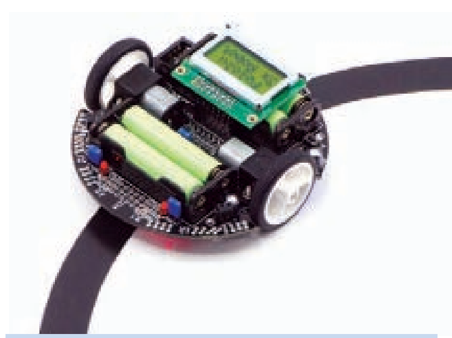


Pololu announces the release of the 3pi robot: a small, high-performance, autonomous robot designed to excel in line following and maze solving competitions. Powered by four AAA batteries (not included) and a unique power system that runs the motors at a regulated 9.25V, 3pi is capable of speeds up to 100 cm/second while making precise turns and spins that don't vary with the battery voltage. This results in highly consistent and repeatable performance of well-tuned code even as the batteries run low.

The robot comes fully assembled with two micro metal gearmotors, five reflectance sensors, an 8x2 character LCD, a buzzer, three user pushbuttons, and more, all connected to a user-programmable AVR microcontroller. The 3pi measures approximately 3.7 inches (9.5 cm) in diameter and weighs 2.9 oz (83 g) without batteries.

The 3pi is based on an Atmel ATmega168 microcontroller running at 20 MHz with 16 KB of Flash program memory and 1KB data memory. The use of the ATmega168 microcontroller makes the 3pi compatible with the popular Arduino development platform. Free C and C++ development tools are also available, and an extensive set of libraries make it simple to interface with all of the integrated hardware. Sample programs are available to show how to use the various 3pi components, as well as how to perform more complex behaviors such as line following and maze solving.

The unit price is $99.95.



For more information, contact:
Pololu Corporation
6000 S. Eastern Ave., Suite 12-D
Las Vegas, NV 89119
Tel: **877-7-POLOLU** or **702-262-6648**
Fax: **702-262-6894**
Email: **www@pololu.com**
Web: **www.pololu.com**

continued on page 93

Complete Development Kits

Renesas — the #1 supplier of microcontrollers in the world — is launching Renesas University, an exciting educational program that gives educators a way to teach microcontroller (MCU) technology using a modern architecture and professional-grade tools. It also offers many valuable resources that help students learn about MCUs and how they can be applied in significant embedded system designs.

#1 MCU
REACH FURTHER

Renesas is a worldwide leader in:

- Microcontrollers
- Embedded flash microcontrollers
- MCUs in car navigation systems
- Power amplifiers for GSM phones
- LCD controllers for color mobile displays

The Renesas University program nurtures an online community where educators and students come together to share ideas, address technical issues and discuss microcontroller topics. It is characterized by:

Publish ▸ Renesas actively encourages academics and students to publish microcontroller-related papers. We provide assistance in publishing course material and microcontroller related books.

Toolchain ▸ The Renesas integrated development environment with toolchain is the commercial version of our development tools – with full C compiler, assembler, linker, and debugger. It is not a typical capability-reduced "educational" version. The only limitation is a 64KB code size after 60 days of use.

Modern ▸ Renesas microcontrollers utilize a modern architecture designed specifically for C and other high-level languages. Our devices handle the most demanding applications of today and tomorrow.

Renesas Starter Kits provide a USB-powered, MCU-based system board with in-circuit debugger/flash memory programmer. They include a CD containing our integrated development environment with toolchain, plus documentation, example firmware and interesting projects.

- **Free for Educators:** Register at the Renesas University website to receive ten free Starter Kits per semester. In return, we request the submission of material that enriches Renesas University; i.e., code, student projects, technical papers, embedded control designs, etc.
- **Low cost for Students:** If actively enrolled in an educational institution, a Starter Kit can be purchased at a very low cost after registering at the Renesas University website.

For more information on Renesas University and how to enroll, please visit **www.renesasuniversity.com** or email: University@rta.renesas.com

Visit us at ESC Boston!
Booth #1008
Embedded Systems CONFERENCE BOSTON

Everywhere you imagine. RENESAS

TEMPERATURE GAUGE PROJECT

BY JIM STEWART

We will build a temperature gauge for the range 0°C to 50°C using a thermistor in an analog circuit. Why use a thermistor? Two words: temperature transducers. Along the way, we will look at various ideas such as how to linearize a thermistor, the effect of thermistor self-heating, and zero and span adjust in a gauge.

Thermistors are inherently non-linear. When using one as a temperature transducer, there are two basic approaches to converting its resistance to the corresponding temperature:

Zero-Power Resistance at 25°C	B Value (°K)	Maximum Power (W)	Heat Dissipation Constant (mW/°C)	Thermal Time Constant (seconds)
10000	4100	0.4	4.5	20

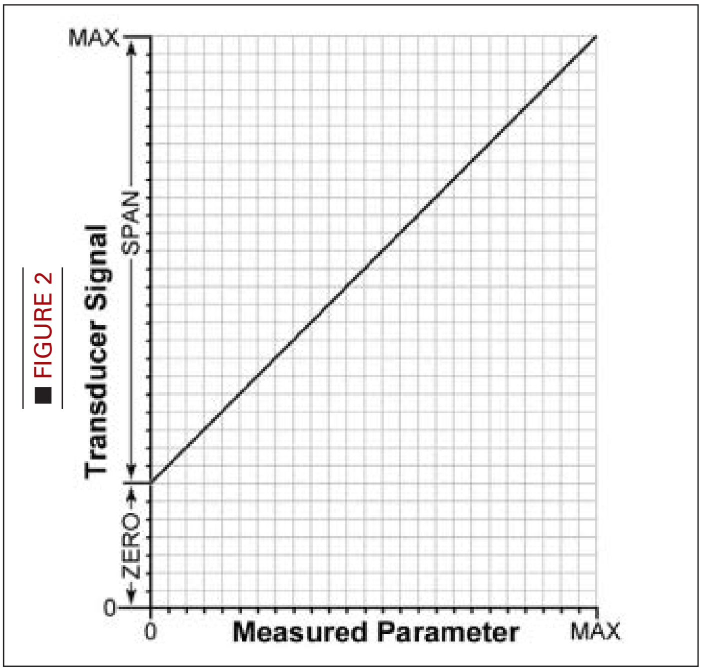
■ FIGURE 1

- **Digital:** Use software and a look-up table to translate resistance to temperature, or solve the equations to get T from the measured R.

- **Analog:** Use a circuit to linearize the resistance vs. temperature characteristic.

A commonly-used equation for thermistor resistance as a function of temperature is:

$$R(T) = R_0 \exp[\ \beta(T^{-1} - T_0^{-1})\ ]$$



■ FIGURE 2

T is the temperature being measured while T_0 is the reference temperature (usually 25°C). R is the measured resistance while R_0 is the resistance at T_0. T, T_0, and the parameter β (Beta) are all in kelvins (K). Beta (sometimes written as B) is a characteristic of the material used to make the thermistor and is given on the device's datasheet.

The thermistor used is Digi-Key part number PNT119-ND. Specs for it are shown in Figure 1. The part was chosen because its disc format makes it more rugged than a smaller bead type. Also, its thermal time-constant of 20 seconds means it will not respond to short puffs of air. A similar device with the same R_0 and β is part # 21T10K from Electronix Express.

The datasheet gives the resistance at 25°C as 10K. We will need the resistances at 50°C and 0°C so we will calculate them here:

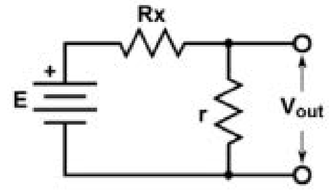
50°C = 323.15 kelvins
25°C = 298.15K
0°C = 273.15K

R @ 50°C = 10K × exp[4100 × (1/323 - 1/298)] = 3.45K
R @ 0°C = 10K × exp[4100 × (1/273 - 1/298)] = 35.2K

Actually, a C language program (RXCALC) was used to do the calculations.

Linearization

A linear function (Figure 2) has the equation y = mx + b where b is the y intercept and m is the slope. Both b and m are constants. In instrumentation terms, b is the *offset*



■ FIGURE 3

while the slope m determines the *span*. A simple way to linearize a thermistor over a limited temperature range is to combine it with a fixed resistor, either in series or in parallel. Figure 3 shows the circuit we will use — a voltage divider. The thermistor is r while Rx is a fixed resistor. With the right value for Rx, Vout will be a reasonably linear function over a limited range of temperature. Figure 4 shows the typical S-curve approximation to a straight line derived from the combination of Rx and r. (The deviation from a straight line is exaggerated). The error is zero at three points as shown. The V vs. T graph would have the same shape. (Calculate Rx the same way for either series or parallel circuits.)

How to Calculate Rx

Referring to Figure 4, we want to select Rx so that (R3 - R2) = (R2 - R1) where:

R3 = Rx || Rtmin (Rtmin = r @ Tmin = r @ 0°C = 35.2K)
R2 = Rx || Rtmid (Rtmid = r @ Tmid = r @ 25°C = 10K)
R1 = Rx || Rtmax (Rtmax = r @ Tmax = r @ 50°C = 3.45K)

After some simple (but tedious) algebra, you get
Rx = A / B where:
A = Rmid × (Rmax + Rmin) - (2 × Rmax × Rmin)
B = Rmax + Rmin - 2 × Rmid

Plugging the numbers into RXCALC we get Rx = 7.67K

We can use a standard 7.5K 5% resistor; the difference between 7.67K and 7.5K is only 170Ω.

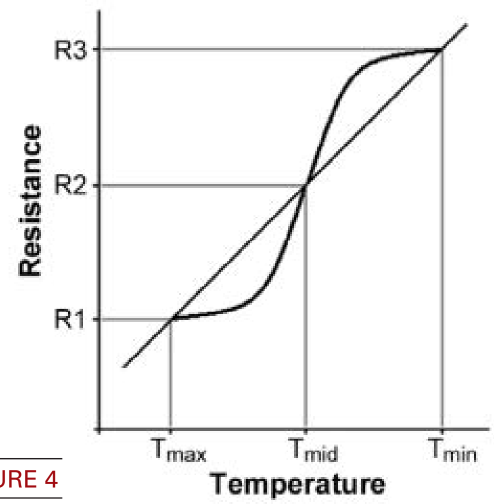
The S-Curve Data

Using the Voltage Divider Rule on Figure 2, we get

Vout = F × E where: F = [r / (R + r)]

T & ΔT (°C)	r (KΩ)	F	Vout	ΔV	ΔV/ΔT
T = 15	16.1	0.684	6.84	---	---
ΔT = (15 - 20)	---	---	---	0.55	0.110
T = 20	12.6	0.629	6.29	---	---
ΔT = (20 - 25)	---	---	---	0.56	**0.112**
T = 25	**10.00**	**0.573**	**5.73**	**---**	**---**
ΔT = (25 - 30)	---	---	---	0.56	**0.112**
T = 30	7.97	0.517	5.17	---	---
ΔT = (30 - 35)	---	---	---	0.55	0.110
T = 35	6.40	0.462	4.62	---	---

■ TABLE 2



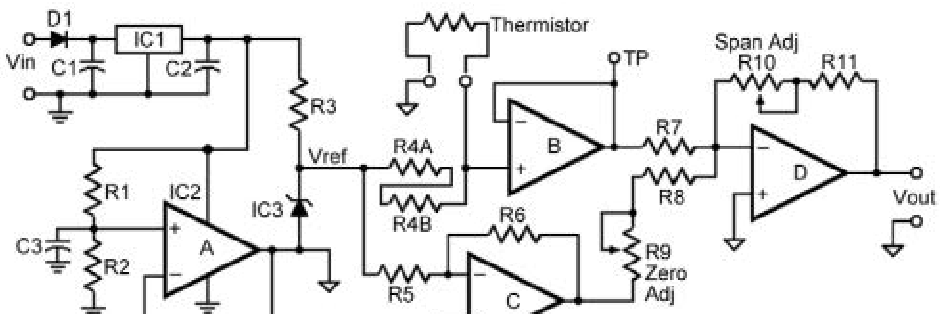
■ FIGURE 4

Table 1 was made with numbers from RXCALC based on E = 10 volts. (The values in the table were rounded off.) In table_1, the slope (DV/DT) from 0° to 5° is 88 mV/°C. As temperature increases, the slope increases to a maximum of 114 mV/°C from 20° to 25°. Past 25°, the slope decreases and is back to 88 mV/°C from 45° to 50°. That sequence of changing slopes generates the s-curve of Figure 4.

Suppose we will use the gauge mostly for temperatures in the range of 20° to 30° with occasional higher or lower readings. Then, the circuit should be as linear as possible in the range around 25°. The previously

T & ΔT (°C)	r (KΩ)	F	Vout	ΔV	ΔV/ΔT
T = 0	35.2	0.821	8.21	---	---
ΔT = (0 - 5)	---	---	---	0.44	0.088
T = 5	26.9	0.777	7.77	---	---
ΔT = (5 - 10)	---	---	---	0.48	0.096
T = 10	20.7	0.729	7.29	---	---
ΔT = (10 - 15)	---	---	---	0.52	0.104
T = 15	16.1	0.677	6.77	---	---
ΔT = (15 - 20)	---	---	---	0.55	0.110
T = 20	12.6	0.622	6.22	---	---
ΔT = (20 - 25)	---	---	---	0.57	**0.114**
T = 25	**10.00**	**0.565**	**5.65**	**---**	**---**
ΔT = (25 - 30)	---	---	---	0.56	**0.112**
T = 30	7.97	0.509	5.09	---	---
ΔT = (30 - 35)	---	---	---	0.55	0.110
T = 35	6.40	0.454	4.54	---	---
ΔT = (35 - 40)	---	---	---	0.52	0.104
T = 40	5.18	0.402	4.02	---	---
ΔT = (40 - 45)	---	---	---	0.48	0.096
T = 45	4.21	0.354	3.54	---	---
ΔT = (45 - 50)	---	---	---	0.44	0.088
T = 50	3.45	0.310	3.10	---	---

■ TABLE 1



■ FIGURE 5

discussed procedure for calculating Rx said nothing about linearity at mid-range.

From Rx to Ry

The slope of a straight line is a constant. In Figure 3, change Rx to Ry. Fix the temperature at 25°, which also fixes r(T), making F a function of Ry. Using calculus, set the second derivative of F with

TEMPERATURE TRANSDUCERS

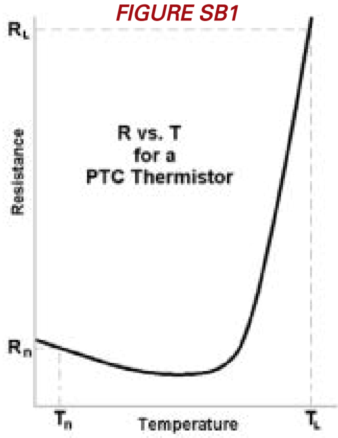
1. Thermistors

A thermistor is a resistor which changes value with temperature. They come in two types: negative temperature coefficient (NTC) and positive temperature coefficient (PTC).

1.1 PTC Devices

Common PTC devices are made of ferroelectric polycrystalline ceramic material. Below its Curie temperature, the material has low resistance; above it, the resistance increases rapidly. Figure SB1 is a typical PTC resistance vs. temperature graph.

FIGURE SB1



Rn is the nominal resistance (e.g., 10Ω) usually specified at 25°C. Current through the device generates heat from the I^2R loss. As temperature increases, resistance at first decreases. At a specified point (e.g., 125°C), resistance increases rapidly to its operating value (e.g., 10K). A specific value of current acts as a "trip-point" past which current becomes limited. PTC devices are overload protectors.

1.2 NTC Devices

Common NTC thermistors are made of a polycrystalline semiconductor such as sintered metal oxide. They're made in shapes such as discs and beads. As the temperature of a semiconductor increases, electrons break free of their atoms to move about and carry charge. As the number of loose electrons increases, it takes less voltage to get current. That is, the resistance drops. Figure SB2 shows R vs. T for a typical NTC thermistor.

NTC thermistors have a non-linear R vs. T curve described by the Steinhart-Hart equation:

$$1/T = a + b\mathrm{Ln}(R) + c\mathrm{Ln}^3(R)$$

where:
T = Temperature in kelvins
R = Resistance at temperature T
a, b, c = Constants obtained experimentally for a given device.

Letting c = 0 in the above yields an equation that is a bit less accurate but a lot easier to use:

$$R(T) = R_0 \exp[\ \beta(T^{-1} - T_0^{-1})\ ]$$

where:
T = Temperature in kelvins
T_0 = Reference temperature (also K)
R = Resistance at temperature T
R_0 = Resistance at temperature T_0
β (Beta) = A constant given on the thermistor datasheet.

Figures SB3 and SB4 are commonly used

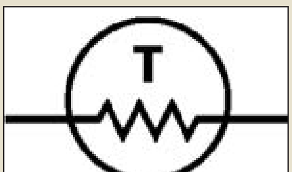


FIGURE SB3

FIGURE SB4

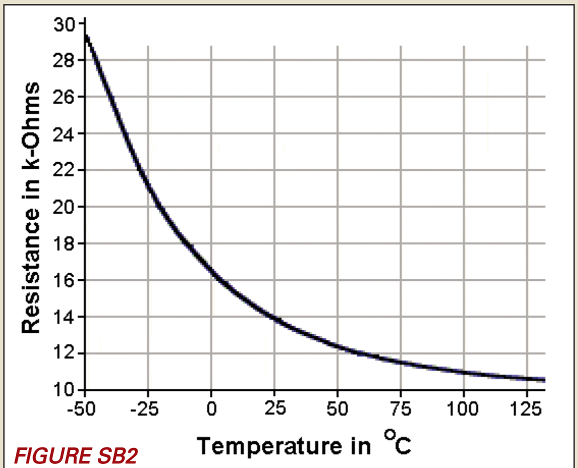


FIGURE SB2

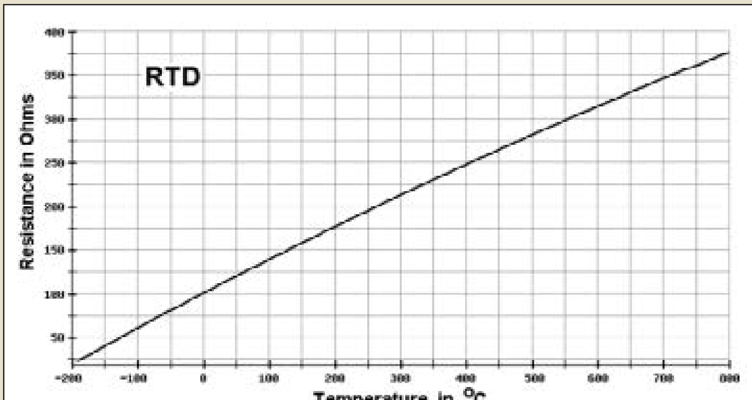


FIGURE SB5

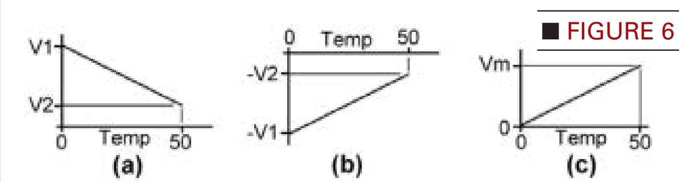
respect to Ry equal to zero and solve for Ry. We'll skip the gory details and give the result:

$$Ry = [(\beta - 2T)/(\beta + 2T)] \times r(T)$$

The program RYCALC (a modified version of RXCALC) will find Ry and the numbers to build Table 2 based on Ry.

In Table 2, we see that $\Delta V/\Delta T = 0.112$ from 20° to 25°, as well as from 25° to 30° while in Table 1, $\Delta V/\Delta T = 0.114$ from 20° to 25° and 0.112 from 25° to 30°. So, there is a small improvement in linearity using Ry.

On the other hand, with Ry the change in Vout from 0° to 25° is 2.52 volts, but from 25° to 50° it's 2.57 volts. Compare that to Table 1, where the respective changes are 2.55 volts and 2.56 volts. That 0.01 volt difference is essentially a round-off error, so Table 1 shows that the three points 0°, 25°, and 50° all fall on the same line as was required. Which is better, Rx or Ry? It's your call.



■ FIGURE 6

Voltage and Self-heating

Voltage across a thermistor generates power as V^2/R, causing heat which changes the resistance. Such *self-heating* causes an error in temperature measurement. The specification *Heat Dissipation Constant* in Figure 1 will be used to estimate the self-heating error.

Self-heating is worst with maximum power in the thermistor. In Figure 3, the thermistor is the load resistance and Rx is the source resistance. According to circuit theory, maximum power in r occurs when r = Rx and the

schematic symbols for thermistors.

2. RTD: Resistance Temperature Device

A typical RTD has fine platinum wire wrapped around a mandrel and is covered with a protective coating. The mandrel and coating are usually glass or ceramic. RTDs have excellent accuracy, stability, and repeatability and are reasonably linear over a wide temperature range. But they are expensive, low resistance, and require careful signal conditioning. Figure SB5 shows R vs. T for a typical RTD.

The resistance of an RTD is given by $R(T) = R_0 (1 + \alpha T)$

where:
T = Temperature in °C
R(T) = Resistance of the RTD at T
R_0 = The resistance of the RTD at 0°C (100Ω is common)
α (alpha) = Slope of R vs. T (ohms per ohm per °C).

The common value for α is 0.00385; another, the so-called American curve, is 0.00392.

3. Thermocouples

Thermocouples are cheap, standardized, interchangeable, and cover a wide range of industrial temperatures. But errors of less than one degree Celsius are difficult to achieve. They're made of two different metals wires joined at one end. The joined end is placed at the point of measurement. By the Seebeck effect, dissimilar metals joined at one end induce an EMF between the wires at the open end. As temperature rises, the EMF rises.

Thermocouples do not measure absolute temperature, but rather a temperature difference. Originally, two junctions were used: one to measure temperature; and the other (the cold junction or CJ) kept at 0°C. See Figure SB6. Modern instruments use electronics to simulate the CJ.

The most common, general-purpose thermocouple is the K-type made from chromel and alumel. They're inexpensive and measure the -200°C to +1,200°C range. They generate approximately 41 μV/°C.

4. PN Junctions

4.1 Diodes

A diode such as the 1N4001 makes a simple but highly linear temperature sensor. With a diode current of about 10 mA, we have:

$$V_D(T) = V_{D0} - \alpha T$$

where:
T = Temperature in °C
α = Approximately 2 mV/°C
V_D = Voltage across the diode
$V_{D0} = V_D$ at T = 0°C.

(Note that the circuit used in this project can be adapted easily to work with a diode.)

4.2 Bandgaps

The silicon bandgap IC is a common temperature sensor. It works by having two matched diodes with different values of I_D. Start with the diode equation:

$$I_D = I_S \times [\exp(qV_D/\eta kT) - 1]$$

where:
I_D = Diode current
I_S = Saturation current ($\approx 1 \times 10^{-12}$ amps)
q = Charge of electron (1.6×10^{-19} coulombs)
V_D = Voltage across the diode
η = "Non-ideal" coefficient ($1 < \eta < 2$)
k = Boltzmann's constant (1.38×10^{-23})
T = Junction temperature in K = 273 + °C

Solve for V_D to get:

$$V_D = K_D T$$

where:

$$K_D = [(\eta k)/q] \times \ln[(I_D + I_S)/I_S].$$

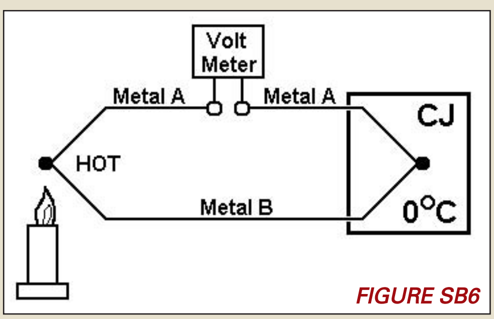


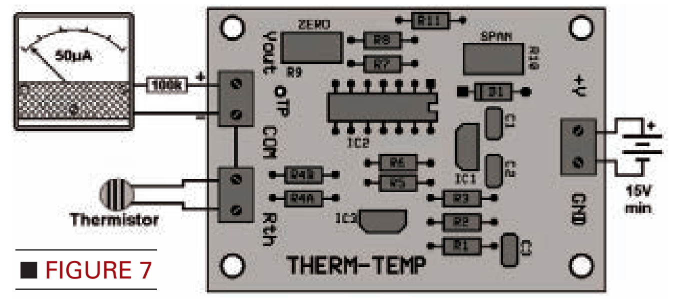
FIGURE SB6

With T held constant, we calculate V_D at two levels of I_D. Let I_H be the higher current and I_L be the lower current. Then, using the above equations, calculate $V_H = V_D$ at I_H and $V_L = V_D$ at I_L. Then, calculate $\Delta V_D = V_H - V_L$. Assuming $I_L >> I_S$, we get $\Delta V_D = (kT/q) \times \ln(I_H / I_L)$. So we can write:

$$T = \beta \times \Delta V_D$$

where:

$$\beta = q / [\, k \ln(I_H / I_L)\,].$$



■ FIGURE 7

voltage across the thermistor is E/2. For E = 10 volts, maximum thermistor power is P = 5^2/7.5K = 3.33 mW. Using the dissipation constant of 4.5 mW/°C gives ΔT = (3.33 mW)/(4.5 mW/°C) = 0.74°C. Suppose we want an error of no more than 0.1°C. To find the maximum usable E, first find the maximum allowable power:

$$P_{MAX} = 4.5 \text{ mW/°C} \times 0.1\text{°C} = 0.45 \text{ mW}$$

The voltage across r to get P_{MAX} is:

$$V_{MAX} = \text{sqrt}(P_{MAX} \times R) = \text{sqrt}(0.45 \text{ mW} \times 7.5\text{K}) = \text{sqrt}(3.375) = 1.84 \text{ Volts}$$

Finally, E = $2V_{MAX}$ = 3.67 volts.

Using E = 2.5 volts from a voltage regulator IC, we will be well within our self-heating limit.

The Circuit

The circuit of Figure 5 provides a 0 to 5 volt output as T goes from 0° to 50° Celsius; a span of 5 volts with a resolution of 100 mV per °C.

IC1 is a voltage regulator. IC2 is a low-power, low-drift, quad op-amp. IC2 is designed to work off a single power-rail, so op-amp (A) is used as a "rail-splitter" to establish a virtual ground above real ground. (The split is not half; we need more "head-room" on the positive side.) IC3 provides a constant 2.5 volts to drive the thermistor.

R4 (R4A + R4B) forms a voltage divider with the thermistor as discussed above. The voltage across the thermistor (Figure 6a) is buffered by unity gain amplifier (B) and appears at TP. With a 2.5V reference, TP ranges from 2.05V (at T = 0°) to 0.77V (at T = 50°).

The buffered thermistor voltage goes to inverting summing amplifier (D). If the thermistor voltage was simply multiplied by -1, it would look like Figure 6b having two problems:

- A zero-offset of negative 2.05 volts.
- A full-scale swing of 1.28V, not the required 5.00V.

To fix the offset, Vref is applied to inverting amplifier

(C). The output of (C) goes to summing amp (D) through R8 and potentiometer R9 — the zero-adjustment. Figure 6c shows the output of the summing amp after the offset is zeroed out. With unity gain, Vm would be 1.28 volts. To get the required 5V span, the gain of (D) is set to about 3.9 by potentiometer R10 — the span adjustment. A nice feature of this design is that the ZERO and SPAN adjustments don't interact.

Construction

The circuit is built on a double-sided printed circuit board. Figure 7 shows the parts layout, along with the connections to power, the thermistor, and an analog display (e.g., an Electronix Express 01DK80 50 μA meter movement). Power can come from two nine-volt batteries, or from a surplus wall mount unit. The connectors are two-position terminal blocks (RadioShack #276-1388).

Calibration

Get an accurate thermometer to measure room temperature along with a bottle of the red insulating paint commonly called "Glyptal." Cut two 12-inch lengths of stranded, insulated hook-up wire. Solder a wire to each leg of the thermistor. Coat the thermistor and exposed wire with the Glyptal (or rubber cement) and let it dry for an hour.

Put a bunch of ice cubes in a bowl and add some water. Wait about five minutes for the bath to stabilize at 0°C. Connect an Ohm-meter to the wires on the thermistor, and insert the thermistor into the ice bath. Wait until the resistance reading stabilizes; it should be about 35.2 kW. Record the actual resistance (call it R_{ICE}).

Attach the thermistor to the Rth terminal block on the board. Adjust R9 (ZERO) until Vout is 0.00 volts.

A printed circuit board for this project can be purchased from the *Nuts & Volts* Webstore @ www.nutsvolts.com or call our order desk, 800 783-4624.

Remove the thermistor from the ice bath and let it warm up to room temperature. Use the thermometer to measure room temperature and record it (call it T_{ROOM}). You can convert Fahrenheit to Celsius using

°C = (°F - 32) × 5/9.

Adjust R10 (SPAN) until Vout matches T_{ROOM}. For example, if T_{ROOM} is 24.3°C, Vout should be 2.43 Volts. (You could use a 50°C hot water bath instead of room temperature. It would give a better calibration, but it's a bit complicated to set up.)

That's it; you should be good to go. Have fun with it. **NV**

PARTS LIST

ITEM	DESCRIPTION
RESISTORS: All 1/4 watt, 5%	
❑ R1	100K
❑ R2	33K
❑ R3	7.5K
❑ R4A	7.5K
❑ R4B	150Ω
❑ R5	100K
❑ R6	100K
❑ R7	10K
❑ R8	10K
❑ R9	5K POT (Bourns type 3296 or equivalent)
❑ R10	10K POT (Bourns type 3296 or equivalent)
❑ R11	33K
CAPACITORS	
❑ C1	0.1 μF Monolithic ceramic, 0.1" spacing
❑ C2	1.0 μF Monolithic ceramic, 0.1" spacing
❑ C3	0.1 μF Monolithic ceramic, 0.1" spacing
SEMICONDUCTORS	
❑ D1	1N4001
❑ IC1	78L12
❑ IC2	TLC27L4BCN
❑ IC3	LM336Z25

HARDWARE
- ❑ Printed circuit board
- ❑ Two-position terminal block: (three required) RadioShack #276-1388 or equivalent
- ❑ Thermistor: DigiKey PNT119 or equivalent
- ❑ Socket: 14-pin DIP (optional)

MICROPROCESSOR CONTROLLED WOOD STOVE

Part 1

BY KERRY BARLOW

Many of us burn wood for the numerous benefits that this fuel may provide. I have been burning wood for many years now and have found that anything that will lighten some of the drudgery from the work involved with this is a distinct benefit.

My particular wood stove is the steel barrel type and is mounted downstairs in my garage. In the past, I was constantly going up and down the stairs all hours of the day to check on the fire. As many of you know who burn wood, this is not a consistent heat temperature fuel. One moment, the fire will be too hot and 20 minutes later, it may be too cool. With these items in mind, I decided to see if there was a way to control the fire's temperature using some sort of a control circuit.

An Atom 24-pin microprocessor was chosen to control the wood stove. The original design called for automatic control of the flue damper (exhaust valve) and combustion door (air input), as well as a boost fan to quickly start a slow fire. I discovered, however, that simply controlling the exhaust damper accomplished everything I wished and more.

■ FIGURE 1

System Operation and Theory

Before I go into any further details, let's look at what the system does and how it operates. The overall design consists of a master control box (Figure 1) and a remote slave display box (Figure 2). The master control box is mounted down in the garage (or basement) and contains the majority of the components necessary for the overall design. Details on construction will be given next month.

The remote slave box is mainly for user feedback upstairs, but it does contain some circuitry not installed in the master control box. Please refer to Schematic 1 which shows the overall system components.

Fire Temperature Control

To control the overall temperature of the fire, I decided to automatically control the exhaust damper as my first goal. I chose a stepper motor (Figure 3) to control the damper for various reasons. Be sure to remember that if the control motor is too close to the fire, the heat may damage it. Limit switches were installed for fully opened and fully closed. (These will be covered in Part 2.)

Temperature measurement is obtained using a MAX6675 thermocouple temperature sensor. I use a K-type thermocouple (Figure 4) with the MAX6675, which provides plenty of heat

■ FIGURE 2

sensitivity range. The MAX6675 reads temperature changes virtually in real time and is well worth the extra effort it entails to use.

I decided upon a semi-closed loop system. I call it "semi" because although the stepper motor does not directly have position output information, I do have feedback coming from the temperature sensor mounted in the flue pipe.

For safety reasons, I also installed a manual shut-off switch to the stepper motor. I had thought this might be necessary in case of problems, however, I never have needed it. In any case, it is there to give me a warm, safe feeling inside!

The software maintains the fire temperature within a user-controlled limit setting. I have found that a 50 degree tolerance range is about the tightest control possible with the fire in the wood stove. Any tighter a range and control problems become insurmountable. From the display panel, you can change these limits while the fire is burning.

The above synopsis seems simple since only three major components are needed. On the software side, it is not that easy because the design entails considerable pitfalls and errors that will develop seemingly on their own. There is a *lot* more to controlling the fire than simply, If temp > high_limit then gosub close_damper.

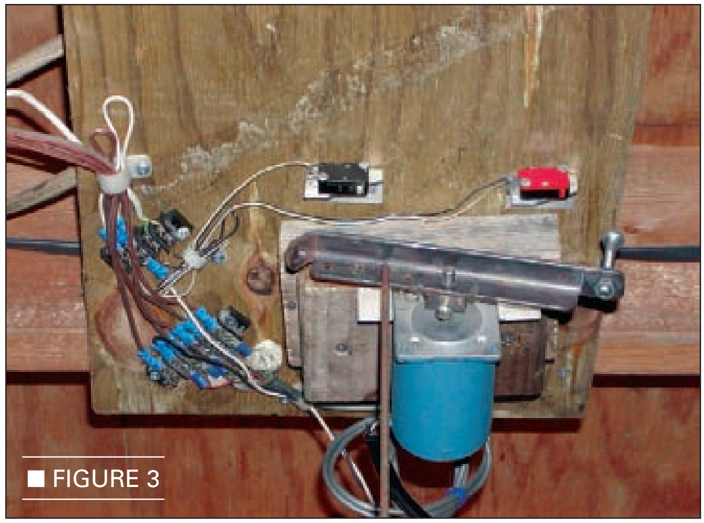
Believe me, I was naive enough to think it would be that easy. I'll enlighten you on these pitfalls as we discuss the software.

Because this is a lengthy project, I will only go into detail regarding the fire temperature control and closed loop feedback software. I will give a brief description of the rest of the program's temperature recording, but I encourage readers to contact me for any further details at mntnweb@echoes.net.

If you refer to the main program woodstove.bas on the *Nuts & Volts* website, you will see that I have laid out the program in sections named as follows: Constants, Variables, Setup code and Setup subroutines, Main program, LCD display subroutines, Exhaust damper control subroutines, Temperature subroutines, Manual switch control subroutines, and Clock subroutines.

These can be broken down into further subroutines. For example, the Temperature subroutine block contains fire, indoor, and outdoor temperature routines. The clock subroutine contains not only clock routines, but also data storage routines. In addition, the program has direct switch control over temperature low and high limit settings. (Please note that all discussion here will take place using the Fahrenheit system.)

As mentioned earlier, for my application I found that a



■ FIGURE 3

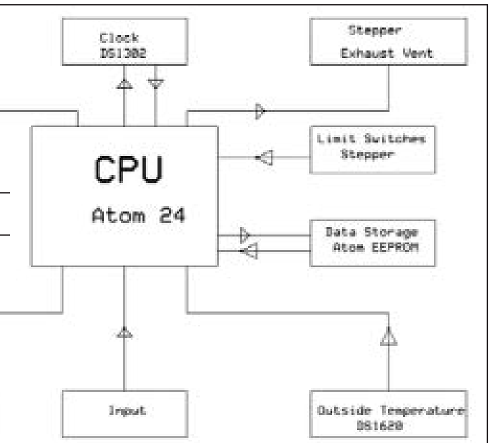
tolerance of 50°F was optimal. Any tighter control and the fire will become unstable. The variables called lowtemp and hightemp are hard-coded in the program, but may be changed using the panel switches in real time while the program is running. This feature allows you to control your overall inside home temperature. If it is too cold inside, simply raise the limits using the manual switches.

Fire Control: Closed Loop Feedback Code

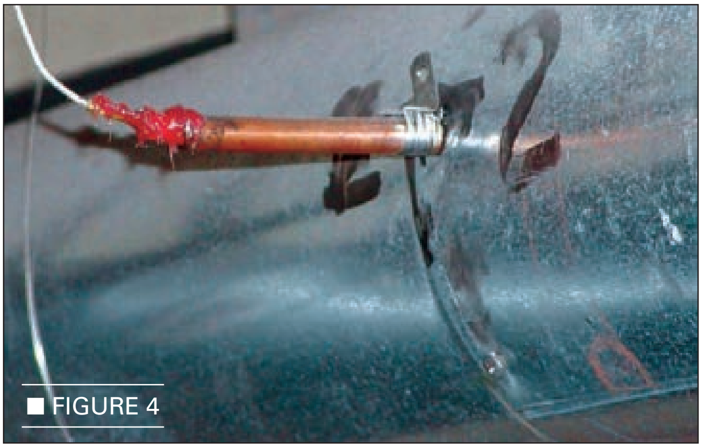
Accurate control of the fire temperature was an involved process and wasn't easy to do. Refer to Schematic 2 for the fire feedback control. The feedback diagram looks simple — and electrically it is — however, the software must perform various functions depending on exactly what the fire is doing in real time. There are 14 different temperature-over-time checks that I make and I change the exhaust damper output based upon them.

(Please refer to the separate program ExhaustDamperControl outines.doc available on the *Nuts & Volts* website. This program is simply the relevant control loop code removed from the main woodstove.bas program so that it is easier for you to read.)

Now let's take a look at the pertinent subroutines and label each of the methods as I come to them. Every single one of these 14 steps is very necessary for their own reasons. One thing to



■ SCHEMATIC 1



■ FIGURE 4

note is that the temphighcheck subroutine is NOT a direct duplicate of the templowcheck subroutine. I discovered that the fire could increase temperature much more quickly than it could reduce the temperature. Additional 'state' checks for the temphighcheck subroutines were necessary. I've labeled the subroutines with intuitive names for easier debugging later on.

FIRELIT subroutine

This innocuous subroutine does a lot more than it appears. When a fire is lit, the program needs to know the position of the exhaust damper (maybe the user has moved it manually since the last fire), so FIRELIT calls the exhauststepcountreset subroutine. This routine drives the exhaust damper to full open, then backs it off four steps so that it is not riding directly on the limit switch. It also updates the LCD display variable stepcount for stepper motor position display.

The subroutine sets a bit called firelit to 1 and increments the fire EEPROM data storage location. In addition (this may not be obvious), at this time when the fire is lit, the IncTime subroutine (accessed from the MAIN loop) calls the IncreaseBurntimeMinutes subroutine, which starts the burn time clock.

FIREOUT subroutine

This subroutine simply resets variables and stores their new value in EEPROM. At this point, the exhaust damper should be in the wide open position because as the fire has been going out, the control software will have driven it to maximum open.

TEMPCHECK subroutine

To effectively control the fire, the present temperature variable called temperature, as well as the old temperature variable (tempold) needs to be saved. The difference between the old and new temperatures is compared over various timeframes and the exhaust damper is changed based on the rate of change of the fire temperature. These various timeframes are incremented with some of the variables listed in the start of this subroutine — AttemptsLow, TempCheckCounter, and AttemptsHigh. I'll explain these shortly. Please note that the TempCheck subroutine is called every 10 seconds (determined by the IncTime subroutine).

1) The first condition tested for in the TempCheck subroutine is the following:

```
If temperature <= lowtemp and temperature > tempold
then goto bypass
```

In this condition, the control loop does nothing. If the temperature is rising on its own, just leave it. There is no reason to force the fire to increase its temperature quickly if it is already doing so on its own at this damper setting. I found that trying to force a quick change only caused grief.

2) The second condition is the opposite of the previous condition:

```
If temperature >= hightemp and
          temperature <= tempold then Goto
bypass
```

Once again, do nothing. If the temperature is falling on its own, then let it be.

3) The third condition in the TEMPCHECK subroutine takes us to the STABLE subroutine (explained momentarily).

4-5) The fourth and fifth conditions take us to the TempLowCheck and TempHighCheck subroutines (covered next).

STABLE subroutine

Don't wait until the fire temperature drops below the lowtemp or goes above the hightemp limit, because this will cause wild oscillations of temperature.

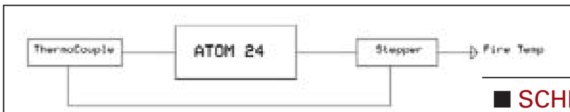
Because we have a 50-degree tolerance level, the fire has a tendency to either run away with itself or cool off way too much before the damper starts responding. I have found (through extensive testing) that if the program waited until the fire temperature fell to the lower set limit, it was too hard to get the fire to increase its temperature. The exhaust damper would open every 10 seconds until it was wide open before the fire would 'catch' and begin burning hotter.

At this point, the fire would race over the hightemp limit and burn too hot. The damper would then try to cool the fire off, thereby causing the same problem, only this time in the cooling mode.

These fluctuations will swing between high and low temperatures as the exhaust damper tries to maintain control. I tried a tighter tolerance value, but this caused other problems. The biggest one was the stepper motor was constantly running open and closed trying to maintain tight control over the fire.

After much trial and error, I determined that the program needs to do 'state' checks on the fire temperature while it is within the lowtemp and hightemp tolerance and then take action accordingly.

The variable tempOldMinute does exactly as its name implies — it stores a 'state' variable once a minute for the following subroutines:



■ SCHEMATIC 2

1) The first condition in the STABLE subroutine is a simple bypass:

```
If temperature = tempold then goto StableBypass
```

Pretty much self-explanatory — bypass around the subroutine if the fire temperature is stable or constant.

2) The second condition is based upon time and temperature:

```
If (tempOld - temperature) >=3 then
    Gosub exhaustopen
```

The fire temperature is falling within the tolerance range, so action must be taken to prevent the oscillations. The temperature has cooled three degrees in 10 seconds, so open the damper one step (remember this subroutine runs every 10 seconds). This is a fast cooling-off period.

3) The third condition is also time based:

```
If (tempOldMinute - temperature) >=6 then
  Gosub exhaustopen
```

The temperature has cooled six degrees in 60 seconds, so open the exhaust damper. Note this loop runs every 10 seconds, so 6 x 10 iterations = 60.

This is a slower cooling-off period, but action is still required. Notice the second state has cooled three degrees in 10 seconds and in this check, the temperature has cooled six degrees in 60 seconds (one degree in 10 seconds).

4-5) The fourth and fifth conditions are the opposites of cooling off — they are for increasing the temperature.

```
If (temperature - tempold) >=3 then
  gosub exhaustclosed
Endif
'Temperature has risen 3 degrees in 10 seconds
' so close damper
If (temperature - tempOldMinute) >=6 then
  Gosub exhaustclosed
Endif
'Temperature has risen 6 degrees in 60 seconds
' so close damper
```

TEMPLOWCHECK subroutine

1) The first state condition is simple, but necessary:

```
If exhaustopenlimitbit = 1 then skip
```

If the exhaust damper is already wide open, skip this subroutine. The damper cannot be opened any further; either the fire is going out intentionally or it needs wood. A buzzer could be added to warn of this condition.

2) The second state condition actually opens the damper one step position, depending on its temperature:

```
If temperature < lowtemp and exhaustmovedopenbit = 0
then gosub exhaustopen
```

I was naive to think this was the only code necessary to control the fire temperature. The exhaustmovedopenbit is used so that the code does not try to open the damper twice within a 10-second period (possibly called by a different IF statement). This line opens the damper in one step increments.

3) The third condition is directly time based:

```
If temperature = tempold and temperature < lowtemp
and AttemptsLow >= 6 and exhaustmovedopenbit = 0 then
AttemptsLow = 0
Gosub exhaustopen
Endif
```

Do the above if Temperature is below lowtemp limit and has also not changed for one minute (six loop iterations x 10 seconds).

It is necessary to raise the temperature if it is too low. It may be out of wood, but until the damper is wide open, we should try opening it one stepper position every minute.

TEMPHIGHCHECK subroutine

Please note! This code is not a direct duplicate of the TempLowCheck subroutine. This is because the fire can increase its temperature much more quickly than decreasing it. Remember, it is more dangerous having a hot fire than a cold one.

1) The first state condition is a safety condition and not used in the templowcheck subroutine:

```
If exhaustopenlimitbit = 1 then gosub exhausttohot
```

The exhaustopenlimitbit is set to 1, showing us the damper is wide open, yet the fire is too hot over the temphigh limit. The subroutine exhausttohot will close the damper 20 steps for a quick reduction in temperature.

```
EXHAUSTTOHOT
     Exhaustopenlimitbit = 0
     For I = 1 to 20
     Gosub exhaustclosed
     Next
Return
```

2) The second state condition is the same as the templowcheck subroutine condition, but is based on a high temperature:

TEMPERATURE SENSORS AND ADDITIONAL SOFTWARE

In the process of my design, I had numerous temperature sensors left over, so I decided to add indoor and outdoor temperatures to a wall-mounted display. Daily outdoor temperature data is stored in the Atom's EEPROM. A DS1302 real time clock was added for temperature over time considerations, as well as data logging of daily temperatures. For a list of all components and exactly what this project consists of, I encourage you to download the separate file "Features.txt" from the *Nuts & Volts* website (**www.nutsvolts.com**). This file will show you all the items my wood stove controller consists of, giving you an idea of what this project entails. Indoor and outdoor temperature readings are not necessary for fire control, so you may consider these as additional features. A real time clock is also not strictly necessary, but it makes things a lot easier. As you will see, many fire temperature decisions are based upon the rate of change over time.

```
If temperature = tempold and AttemptsHigh >=6 and in4
= 0 then
AttemptsHigh = 0
Gosub exhaustclosed
Endif
```

3) The third state condition is also different from templowcheck:

```
If temperature > tempold and AttemptsHigh >=4 then
Exhaustclosequantity = temperature - tempold
Gosub heatingfast
Endif
```

The fire is in a fast heating condition. We have been in this loop four times now trying to close the damper one step at a time and it has not reduced the fire temperature adequately (40 seconds has passed). The HeatingFast subroutine will take care of this for us.

HEATINGFAST subroutine

Called here by the temphighcheck subroutine, we have a fast heating fire. This subroutine will close the damper a variable amount based upon how quickly the fire temperature is rising:

```
HeatingFast
      AttemptsHigh = 0
       If Exhaustclosequantity >=4 then
       Exhaustclosequantity = 4
       Endif
       AttemptsHigh = 0
       FOR I = 0 to Exhaustclosequantity
       Gosub exhaustclosed
       Next
Return
```

Notice the For-Next loop. Exhaustclosequantity contains a value based upon the temperature change (as calculated in the temphighcheck subroutine we just left). Close the damper this amount and reset the AttemptsHigh counter (because 40 seconds have passed).

Additional Software

Temperature Sensor Software: Indoor and Outdoor

Temperature measurement is fairly straightforward for inside measurements, however getting the outdoor DS1620 sensor to display proper temperature values below 32°F was a major headache. To my knowledge, I have written the only working temperature code for the DS1620 and the Atom microprocessor. I have broken the code up into subroutines above 0 degrees and below 0 based upon the DS1620 output bit's sign. Note that the NegTemp variable is assigned a byte variable, even though the DsData output is in word format. I did this to strip the extra eight bits from the DsData word variable. I encourage you to review this code closely if you wish to use it. I will be happy to answer any further questions readers may have.

DS1302 Clock Software

A separate program DS1302_ClockProgram.bas is used for initial setup of the clock chip. This was done to reduce the size of the main program. It is only needed one time.

The woodstove.bas software is fairly straightforward. The DS1302 outputs data in BCD format. The LCD display and EEPROM use decimal format, so a conversion is necessary. The following formula works well with the Basic Atom:

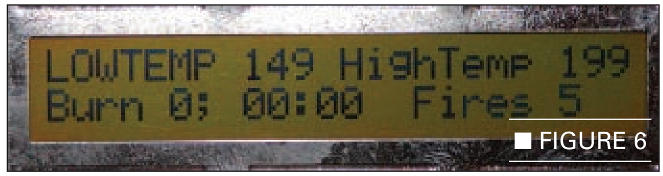
DecimalValue = (BCDValue.nib1 * 10) + BCDValue.NIB0

This conversion formula is used in the UPDATEOUTSIDE TEMP subroutine. The basic Atom compiler automatically converts into decimal when using the proper syntax inside the serout command.

Datalog in EEPROM

The Atom has 256 bytes of internal EEPROM storage memory. Daily low and high temperatures are stored inside the Atom. Positive numbers only can be recorded in EEPROM, so I convert negative temperatures to positive simply by adding a

■ FIGURE 5



■ FIGURE 6

■ FIGURE 7

PARTS LIST

ITEM	SUPPLIER WEBSITE	PART #
❑ Atom 24-pin CPU	http://BasicMicro.com	B0017
❑ Dallas Maxim ICs	www.maxim-ic.com	DS1302, DS1620, Max6675
❑ Max6675	www.jameco.com	884766
❑ DS1302	www.jameco.com	176778
❑ DS1620	www.jameco.com	114382
❑ HeatsinkT0-3	www.jameco.com	151546
❑ Crystal	www.parallax.com	251-03230
❑ SOIC Adapter	http://epboard.com	E8-0003
❑ Thermocouple	www.virtualvillage.com	TP-01
❑ LCD 24 x 2	www.allelectronics.com	LCD-107
❑ Stepper	www.motors.wrobots.com	Various models available
❑ Stepper IC	www.elabinc.com	EDE1200
❑ Protoboard	http://unicornelex.com	32-3004

value of 100 to every temperature reading and then storing the converted temperature. When the FIRES subroutine reads from EEPROM, I subtract 100 to get the proper value to display. See Figure 5 for an example of the display's output.

LCD Display Code

There are five different display modes that may be selected with the manual switches.

The default LCD mode shows: inside, outside, and fire temperatures; time and date; exhaust vent stepper position and what mode the vent may be in; opening, closing, or stable (refer back to Figure 2). Note in the figures there are two different lines of text for the buttons. The upper line is for the temperature mode and the lower line is for the clock setting mode.

Pushing the right hand Daily Log switch accesses the second mode. This mode shows yearly low and high temperatures, as well as daily low and high temperatures (Figure 5).

Pushing the left or right hand lower and raise buttons will select the fire temperature limit display and show the fire's current burning time and how many fires you have lit this year (Figure 6).

Holding either the lower or raise button down for more than three seconds will enable the temperature setting subroutine. The subroutines RaiseTempLimit and LowerTempLimit will change the low and high limit temperatures, and thereby change the display output, as well.

Manual Switch Code

Holding down the right hand Set & Hold switch and either one of the Hours or Minutes switches accesses the final program mode. This mode will not be used very often; it is designed to set the daily time of the clock if it should ever lose accuracy (Figure 7).

Setup Code

Initial program setup and runtime is also straightforward. Variables and constants are declared and then setup subroutines are called. As mentioned earlier, the exhaust damper position must be set and — in case something drastic has happened and all EEPROM values have been corrupted — the setup routine will input initial EEPROM values for temperatures.

This was simply done because with all the constant code changes I was making, I noticed that sometimes disconnecting power or program restarts might corrupt the EEPROM data.

Wrap-Up

As you can see, controlling the temperature of a wood fire is not as simple as it might appear. This project is quite involved and I have only touched upon the majority of the software code. In the next article, I will go into details of hardware design, schematics, and parts sources for actual construction. **NV**

M16C/64 & 65 - high performance additions to the M16C Platform

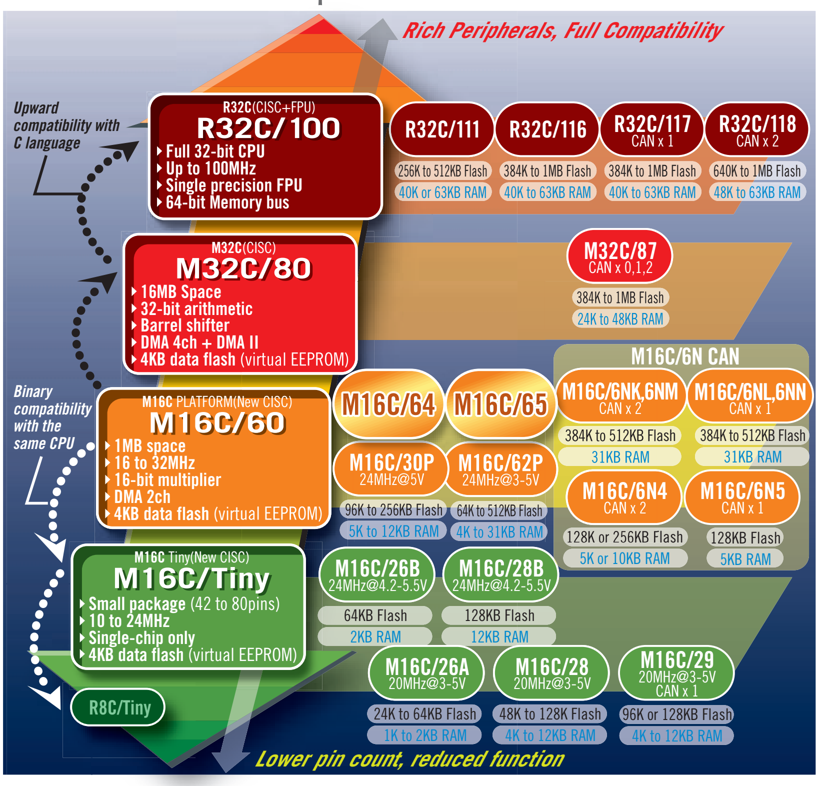
Up to 768K Bytes Flash, up to 47K Bytes RAM in 80- to 128-pin package

Renesas Technology

No. 1* supplier of microcontrollers in the world

presents the M16C/64 and 65 additions to the M16C Family, the industry's only fully code compatible platform that addresses the entire 8-bit through 32-bit price/performance application space. The M16C Family of devices protects your code investment with seamless code and pin compatibility across the platform. The M16C Platform features optimized peripheral mapping and on-chip memory that permits applications ranging from 24K Bytes to 1M Bytes, with between 42 and 144 pins, in an efficient 16-bit CISC architecture. Reduce your overall development time and cost with a cross-platform code base and development tools from Renesas Technology.

M16C Platform Product Lineup

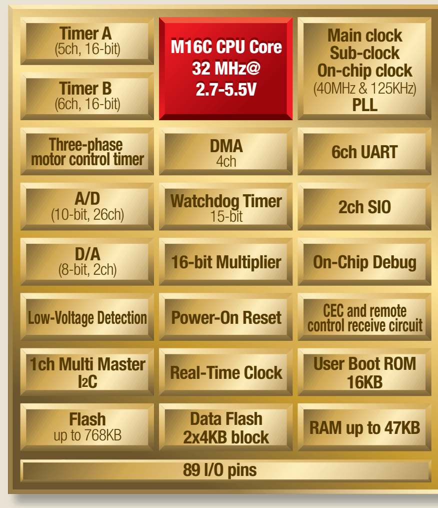


HOT Products

M16C/65 Group

"Setting a new standard for mid-range devices"

M16C/65 Block Diagram (*M16C/65 100pin example)



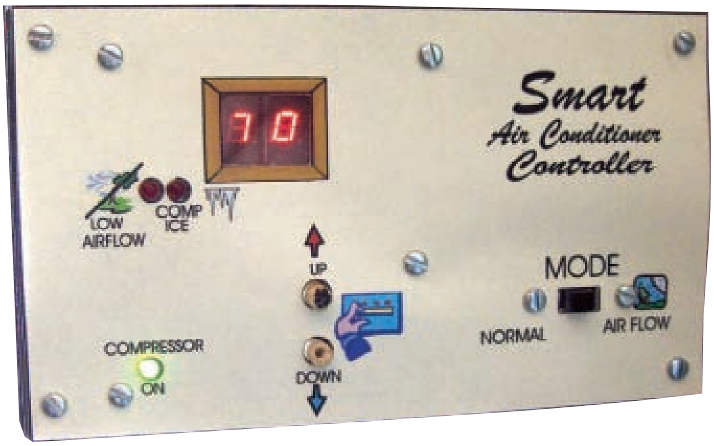
Top Reasons to Select M16C/65

- **Broad Platform compatible to M16C/62P**
  - 128KB - 768KB Flash; 12KB - 47KB RAM; 80 - 128 pins
- **System cost down, through integration**
  - 40MHz On-Chip Oscillator
  - 125KHz On-Chip Oscillator for Watchdog
  - Power-On Reset circuit (POR)
  - Low-Voltage Detection (LVD)
  - Real-Time Clock (RTC)
- **Powerful 32MHz CISC operation**
  - High-speed hardware multiplier
  - 4 Direct-Memory-Access control channels (DMA)
- **Extensive communication interfaces**
  - 6ch scalable async/sync serial I/O channels
  - 2ch synchronized channels additional
  - 1ch I2C master interface
- **Other features**
  - 10-bit A/D with 1.72us conversion
  - 2ch 8-bit DAC output
  - 11ch 16-bit timer
  - Three-phase motor control unit

*Source: Market research firm iSuppli Corp. (El Segundo, Calif.)

MAKE THE SMART ROOM AIR CONDITIONER CONTROLLER

BY TOM FOX



A Microcontroller Based Multipurpose A/C Remote Thermostat

This PICAXE project was originally designed to control a room A/C for use in an inexpensive walk-in cooler. However, this is just one of the duties this smart gadget is capable of.

Most practical ideas come from a real-life problem that needs to be solved. The problem that inspired this project was the need for a walk-in cooler that I could afford, which could be kept at the temperature of an average fridge (38°F to 40°F). I already had an old room A/C and thought there was a possibility I could use it as the refrigeration unit. The problems in using an inexpensive room A/C are legion and, at first look, a bit frightening.

First, the obvious one. The thermostat in most room A/Cs will only go down to the low to upper 60s. Second, unless you live in parts of Nevada or surrounding areas where the dewpoint is often below 30°F, the evaporator coil will ice up if you simply replace the A/C's thermostat with one that functions down below 45°F. This not only will keep the A/C from cooling but could damage the compressor!

Third, if you are going to control the compressor directly by replacing the thermostat, you need a compressor protection control so there is at least a three minute delay before the compressor restarts. The last problem that needed to be solved (and this is the one that makes the project worthwhile for the average reader) has to do with a remote reading thermostat with digital accuracy. With this project, you can set the exact temperature you want from any place in the room you want — even from your bed!

The Basic Design

This project is ideally suited to the use of a microcontroller (there are actually two used here). I remember commenting in my book on the HC11 (*Programming and Customizing the HC11 Microcontroller*) that one shouldn't bother using a microcontroller in designs such as a simple thermostat. However, the PICAXE changed my opinion since it is so simple, inexpensive, and reliable to use in such designs. It is also mercury-free! While the thermostat in the Smart A/C Controller has far more capabilities than most, its design is surprisingly simple — thanks to the PICAXE.

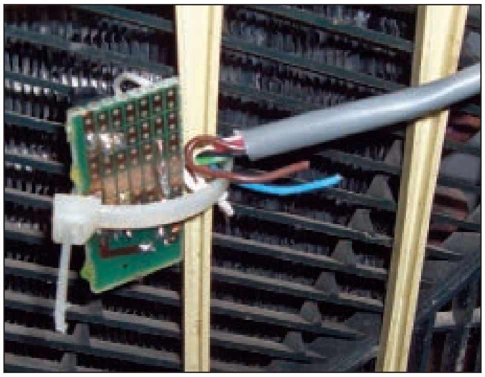
The design here uses two PICAXEs — the 28X1 and an 8M. Most of the smarts come from the 28X1. The 8M's primary purpose is rather mundane — it functions as a safety since it assures that there is a three or five (default) minute delay between the time the compressor shuts off and it turns on again.

Modifications Needed to a Room Air Conditioner

There are two ways to control the room A/C. The first way doesn't require you to actually modify the A/C. What you do here is fool its thermostat! To do this, you first set the A/C's built-in thermostat as cool as it will go if you want

to use the Smart thermostat in a walk-in cooler or as warm as it will go if you simply want to use it as a remote digital thermostat for a room. Next, place a small light bulb right next to the thermostat. Then, when the Smart thermostat calls for cool the light bulb lights and the room A/C thermostat thinks it's too warm and turns on the compressor.

The second way (which is what I did) is to replace the A/C's thermostat with a power relay.



■ Close-up of the air flow sensor assembly mounted in the A/C's air stream.

■ The Smart A/C Controller mounted on a wall. Also shown is the A/C and a junction box containing the transformer.

Smart Thermostat's Design Details

This project uses the National Semiconductor's LM34 chip as temperature and airflow sensors. The LM34 — a three terminal device — is especially easy to use with the PICAXE. Connect +5 volts (up to 20 volts) to the LM34's Vs terminal, ground to GND, and you get 10 mV per degree Fahrenheit at the LM34's Vout — all with one degree (with the LM34A or LM34CA) accuracy! Of course, with a single power supply you can't measure below zero if you reference the output voltage to ground but with most applications, that's not a problem.

The airflow sensor makes use of the fact that a heated object is cooled when air flows over it. Here, we use an unheated LM34 next to a LM34 heated with a 100 ohm resistor. With no airflow, the heated LM34 is roughly 25 degrees warmer than the unheated one. With strong airflow, there is just a couple degrees difference. The PICAXE 28X1 firmware keeps an eye on this difference and if it is too high, it shuts off the compressor since most likely ice has built up. The ice then slowly melts since the fan is still going. After it melts, the compressor turns on and everything — hopefully — will be copasetic.

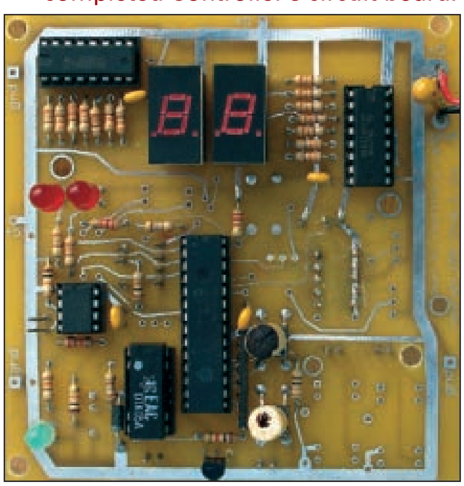
Once you have the information, you have to do something with it. Here, the PICAXE 28X1 is used to make sense of the information and issue all the commands to the A/C. The PICAXE 28X1 is basically a Microchip PIC16F886 microcontroller with a sophisticated Basic interpreter built into ROM. For more on the PICAXE, see back issues of *Nuts & Volts*. This PICAXE has many features ideally suited to this project and it is so simple to use if you are at all acquainted with the Basic language.

The display consists of two seven-segment common anode LED displays, each controlled by a 75HC595 eight-bit shift register with output latches. The firmware in the PICAXE translates data into codes sent to the shift registers which then lights the LED's segments. (See Listing 1 which is available on the *Nuts & Volts* website at **www.nutsvolts.com**.)

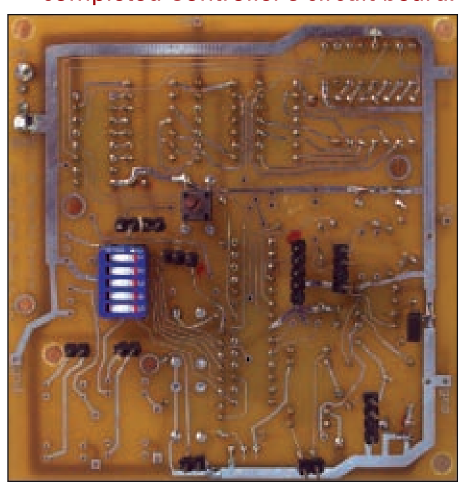
The primary purpose of this project is to intelligently control a compressor. Let us assume you decide (like I did) to control the compressor directly. To do this, you must replace the thermostat with a power relay with contacts rated at least at 15 amps at 120 volts AC. For the coil, I chose a relay with 24 VAC coil since this is the type normally used to control heating and cooling equipment. Check the A/C's power rating. If you want to go the indirect root by heating the A/C's built-in thermostat, you will still need a suitable relay to control the light bulb (or other small heater).

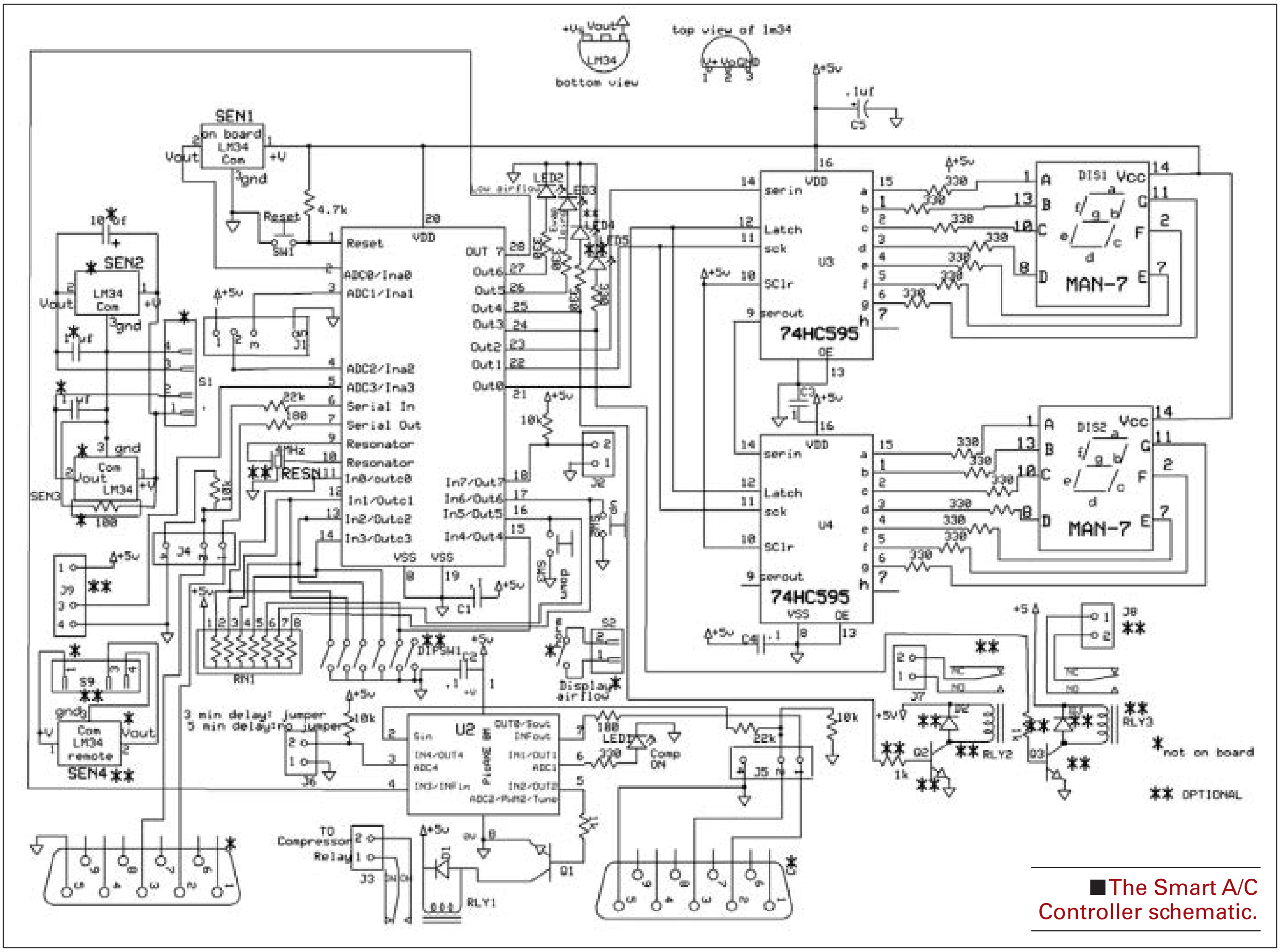
Now getting back to our circuit. The coil of the power relay is connected (in series with a power source) to the N.O. contacts of RLY1, via connector J3. This relay is controlled by U2, which is an 8M microcontroller. Notice that the PICAXE 8M microcontroller's Input 3 pin (terminal 4) monitors the 28X1 Output 7 pin (terminal 28). The 8M program (see Listing 2 also on the *Nuts & Volts* website) makes sure that it doesn't turn on the compressor unless it has been off for five (or three) minutes. LED1 is connected to the 8M's terminal 6 which is set up as an output. This LED starts flashing immediately

■ Close-up of the component side of the completed Controller's circuit board.



■ Close-up of the foil side of the completed Controller's circuit board.





■ The Smart A/C Controller schematic.

after the 8M receives a "compressor on" order from the 28X1. It then will go on steadily after the delay time is achieved *and* as long as the 8M continues to receive a compressor on order. At this same time, the 8M issues a signal to activate RLY1.

The reader may notice there are several parts listed, such as RLY2, RLY3, LED4, LED5, RESN, DIPSW1, and SEN4, which are shown on the schematic (and circuit board) but are optional. These parts were designed into the circuit board for future expanded capabilities of the controller.

Making It

The prototypes of the Smart Controller use professionally made, double-sided printed circuit boards (PCBs) with plated through holes, but without silk-screening. These boards were made by **expresspcb.com**. I also took advantage of their free schematic and layout software, both of which I can recommend.

While most parts in this design are standard size with .1" pin spacing, if you use the foil pattern provided with the files on the website (or purchase a PCB through **www.magiclandelectronics.com**), make sure you either use the exact pushbutton switches called for in the parts list or only substitute switches that have nearly identical terminal placements.

I recommend that the reset switch, the optional DIP switch, and all the connectors (J1, J2, etc.) be mounted on the foil side and the other components on the normal component side.

If your particular case has good air circulation, SEN1 (room sensor) can be mounted directly to the board. If not, it is best to attach a short three-conductor cable to it and place the sensor outside the case, like I did.

The air flow sensor is constructed using a tiny PCB. Make sure the 100 ohm resistor (heater) contacts the sensor. A tiny bit of epoxy cement can be used here.

Increased Accuracy

For higher potential accuracy, you can tweak the voltage regulator so it puts out exactly 5.12 volts. Alternatively, you can modify the program a bit. For instance, if the supply voltage is exactly five volts, you will want to multiply the average reading you get from the 10-bit A/D converter by 5,000 and then divide the result by 5,120. The result is the binary number of the temperature (in half degrees Fahrenheit). In general, multiply the average reading by the EXACT supply voltage in millivolts and then divide this by 5,120.

The power supply for the controller is a standard five volt one. However, the 28X1 program treats the voltage as if it was exactly 5.12 volts. To simplify things, I assumed a voltage reference of 5.12 volts and then treated each .005 volts as .5 degrees Fahrenheit (5.12 divided by 1024 is .005). I found that most five volt regulators produce an output voltage of somewhere between 5.05 and 5.10 volts. Thus, a 5.07 volt regulator should, theoretically, be 1% accurate.

For the power supply, I purchased a 24 VAC transformer at a hardware store. This transformer also has 8 VAC and 16 VAC outputs. I used the 8 VAC output to power my +5 volt power supply and the 24 VAC output to power the compressor's relay.

Programming the PICAXEs

The programming port for the 28X1 is J4 and the port for the 8M is J5. If your computer has an RS-232 serial

Other parts needed to complete your project include: 5V power supply, 24 VAC transformer along with a power relay with a 24 VAC coil (with a resistance of over 70 ohms), connectors, wire, case, and, of course, a room air conditioner.

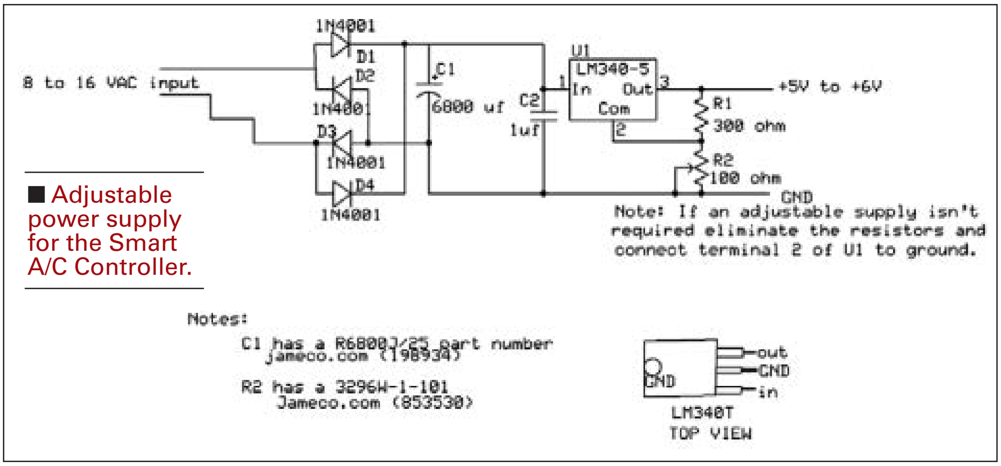
PARTS LIST FOR CONTROLLER BOARD

(Note: Optional parts and parts not mounted on the circuit board are not listed here.)

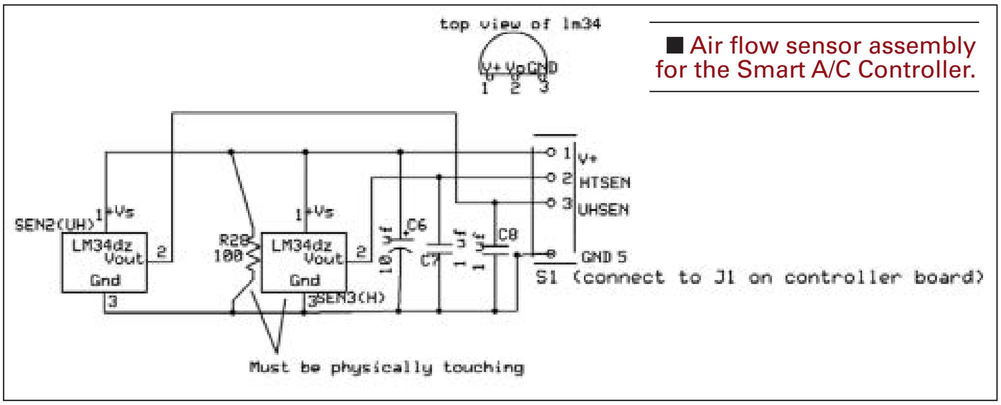
QTY	ITEM	DESCRIPTION	PART #
Semiconductors			
❑ 1	D1	1N4001 silicon rectifier	1N4001
❑ 1	Q1	NPN transistor, TO-92	PN2222A
❑ 1	SEN1	LM34 temperature sensor	LM34DZ
❑ 1	U1	28X1 PICAXE	PICAXE-28X1
❑ 1	U2	8M PICAXE	PICAXE-8M
❑ 2	U3,U4	74HC595 eight-bit shift reg/out latch	74HC595N
LEDs			
❑ 2	DIS1,DIS2	.3" seven-segment common anode LED display (MAN-7 EQUIV)	UA3051-12 or #334984 (**Jameco.com**)
❑ 1	LED1	Green T1-3/4 LED	LG13740 or #334086 (**Jameco.com**)
❑ 2	LED2,LED3	Red T1-3/4 LED	UT1871-81-M1 or #333973 (**Jameco.com**)
Resistors			
❑ 17	R1-R17	330 ohm, 1/4W	1/4W, 330 OHM
❑ 1	R18	4.7K, 1/4W	1/4W, 4.7K
❑ 1	R19	100 ohm, 1/4W	1/4W, 100 OHM
❑ 2	R20,R21	22K, 1/4W	1/4W, 22K
❑ 2	R22, R23	180 ohm, 1/4W	1/4W, 180 OHM
❑ 4	R24-R27	10K, 1/4W	1/4W, 10K
❑ 1	R28	1K, 1/4W	1/4W, 1K
❑ 1	RN1	10-pin 10K in-line resistor network	4610X-101-103LF #773505 (**Jameco.com**)
Capacitors			
❑ 4	C1-C4	.1 µF 50V capacitor	SR215E334MAA or #544930 (**Jameco.com**)
❑ 1	C5	10 µF 16V tantalum capacitor	TM10/16 or #94062 (**Jameco.com**)
Hardware			
❑ 1		28-pin DIP socket	
❑ 1		Eight-pin DIP socket	
❑ 3		14-pin DIP sockets	
❑ 2		16-pin DIP sockets	
❑ 1	J1	Five-pin .100" male header (remove pin 4 and put in polarizing peg)	22-23-2051 or #232291 (**Jameco.com**)
❑ 3	J2,J3,J6	Two-pin .100" male header	22-28-4020 or #879406 (**Jameco.com**)
❑ 2	J4,J5	Four-pin .100" male header	22-23-2041 or #232282 (**Jameco.com**)
❑ 3	SW1-SW3	Tactile switch with .17"x.25" lead spacing	KT11P4CM or #149948 (**Jameco.com**)
Miscellaneous			
❑ 1	RLY1	5V DIP relay	201A05 or #106463 (**Jameco.com**)

PARTS LIST FOR AIR FLOW BOARD

❑ 2	SEN2,SEN3	LM34 temperature sensor	LM34DZ
❑ 1	R28	100 ohm 1/4W resistor	
❑ 1	C6	10 uf 16V tantalum capacitor	TM10/16 or #94062 (**Jameco.com**)
❑ 2	C7,C8	1 µF 50V ceramic capacitor	CK05BX105K or #1776185 (**Jameco.com**)
❑ 1	Printed circuit board (See foil pattern files to make your own. A double-sided board with silk-screening is available from Magicland Electronics. See **www.magiclandelectronics.com** for details.)		



■ Adjustable power supply for the Smart A/C Controller.



■ Air flow sensor assembly for the Smart A/C Controller.

designed into many other projects.

To review, Listing 1 is the source code for the PICAXE-28X1 and Listing 2 is for the PICAXE-8M. Of course, you will also need PICAXE programming software to download the firmware into the PICAXEs. Happily, this software is available as a free download from **www.picaxe.co.uk**. In addition to the free software, you can obtain an enormous amount of easy-to-understand information on the PICAXE from this website.

Installation

This controller can manage the A/C's compressor directly by replacing the A/C's thermostat with a power relay, or it can be used to turn on a small light bulb located next to the A/C's thermostat. Installation details are left to the reader. Keep in mind that the two pin connector J3 acts as an on and off switch with a 1/2 ampere rating. It is the connector to use to control either the compressor's relay or a light bulb relay.

Even if you don't plan on using this A/C controller in a walk-in cooler, it is wise to still place the air flow sensor board right next to the A/C's evaporator coils so it can sense the air flow. This is especially true in humid climates when the thermostat setting is below 70 degrees. Make sure you

interface, connect pin 1 of J4 or J5 to pin 2 of a nine pin D female socket; pin 2 of J4 or J5 to pin 3 of the nine pin D female socket; and pin 4 of J4 or J5 (the ground pin) to pin 5 of the D female socket. If your computer only has a USB port, you will need a USB-to-serial adapter. These adapters — which usually come along with the necessary drivers on a CD-ROM — are now quite inexpensive, easy-to-use, and can be

put a coating of waterproof sealant over the air flow assembly's foil, exposed wires, connections, and leads.

After you determine the method you are going to use to control the A/C, mount the Smart A/C Controller in a convenient location.

Using It

With the MODE switch in the NORMAL position, the display shows the temperature of the room.

When power is first applied, the green *Compressor On* LED will flash twice. Then, the red *Low Airflow* LED will flash once, as will the *Freeze Up* LED. This sequence indicates that both MCUs are operating properly. Next, the display will show the temperature in degrees Fahrenheit. (If you want to display Celsius, install optional DIPSW1 and put position 1 ON.) The default compressor delay is five minutes. If a jumper is placed at J6, the delay is three minutes.

You can find the temperature of the thermostat setting by momentarily pressing either the UP or DOWN buttons. The setting will flash for 10 seconds. While the display is flashing, you can set the temperature using the UP and DOWN buttons.

With the MODE switch is in the

Tips on Setting up a Walk-in Cooler

As mentioned earlier, the Smart A/C Controller's original purpose was for use in a walk-in cooler that used an inexpensive room air conditioner as a cooling source. If you plan on using this project for this purpose, I have a few tips for you. If you use an average sized air conditioner, the room should be less than 1/2 the size of the rating of the air conditioner and it must be well insulated. The best way to do this is to use foam insulating sheeting to partition off a room. This partitioning job is, in fact, quite easily done. The primary problem is the door. I used a combination sliding insulating door and clear plastic vertical strips. Another tip: Don't set the thermostat below 38°F. You want to start out at 45°F.

Hint: A small fan blowing on the A/C's evaporator coil may help reduce ice build-up and thus could make the A/C more efficient.

AIR FLOW position, the display shows half degree differences between the heated and unheated sensors in the air flow module. When the air is dead calm, this reading often reaches 45. With strong air flow, it drops below 15. The program keeps an eye on this air flow number and when it rises above 18 (the default setting), it shuts off the compressor. The procedure to change this setting is similar to changing the thermostat's SET temperature.

Enhancing and Expanding the Smart A/C Controller

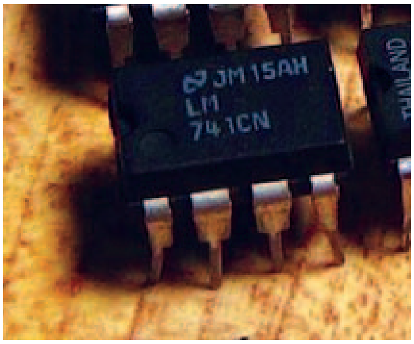
This project has enormous expansion capabilities. The optional parts RLY2, RLY3, SEN4, LED4, LED5, S6, RESN, DIPSW1, and associated parts are shown in the schematic. The circuit board was designed to include these parts. However, except for DIPSW1, the firmware in the 28X1

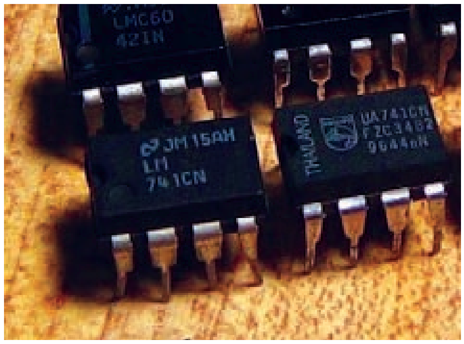
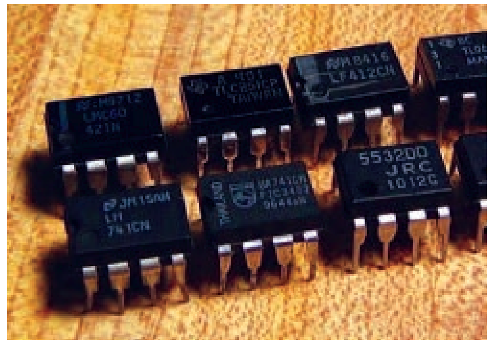
does not know they exist! Nonetheless, with the use of J4 and/or J5, a computer, the PICAXE programming software, and a bit of know-how the reader can re-program the PICAXE for their own unique application.

One possibility is to include controls for a heater so that the room temperature doesn't drop below freezing. Another possibility is to use the remote reading sensor to keep track of outside temperature for a more even and comfortable indoor temperature. Since the 28X1 can handle a 4,096 byte program and only 710 is used in the initial design, there is plenty of room for ingenuity! **NV**

CONTACT THE AUTHOR

Tom Fox can be reached at magiclandelectronics@gmail.com or at his websites at ***www.magiclandelectronics.com*** *and* ***www.magiclandfarms.com****.*



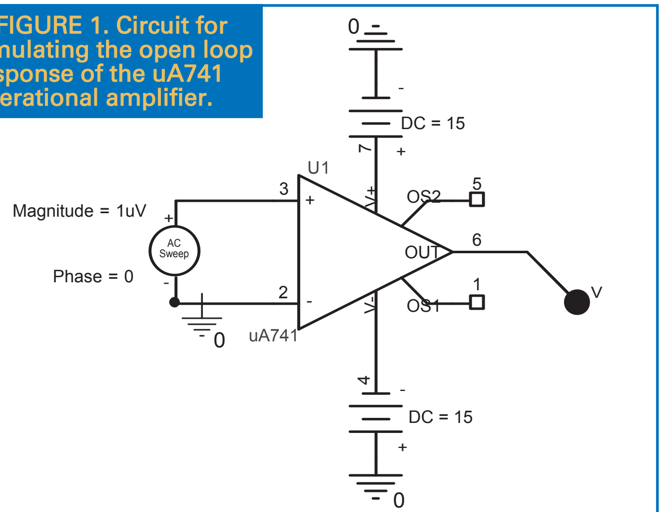



Inverting and Non-inverting Amplifier Design Using OP-AMPS

This article discusses simulation results for both inverting and non-inverting amplifiers, using the uA741 op-amp. Additionally, the article highlights how overall amplifier performance is affected by the gain and bandwidth limitations of the op-amp, as illustrated by PSpice simulations. The uA741 operational amplifier is discussed because it is included in most versions of PSpice, so you can easily repeat the simulations on your own. However, the theory presented here applies to any op-amp used with feedback. We'll also discuss a simple audio amplifier that you can build in order to validate the concepts discussed here.

by Justin Luhm and John E. Post

■ FIGURE 1. Circuit for simulating the open loop response of the uA741 operational amplifier.



Prior to the development of digital computers, complex mathematical equations were solved using analog computers. An analog computer is formed by combining individual amplifiers wired to perform the mathematical operations of addition, subtraction, integration, and differentiation. For this reason, the amplifiers used in analog computers became known as operational amplifiers or op-amps.

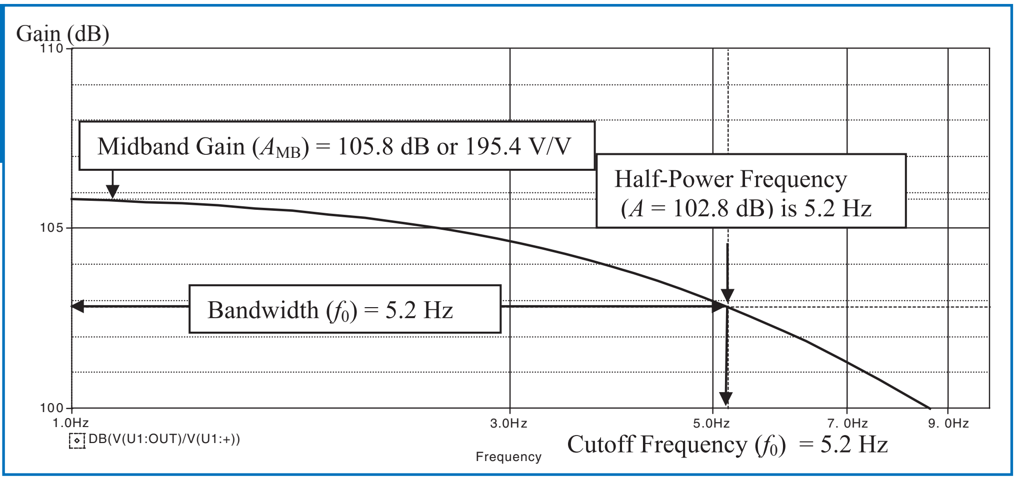
Operational amplifiers were initially constructed using vacuum tubes and then transistors, usually as individual circuit boards that were wired together. With the advent of integrated circuit technology, op-amps were inexpensively mass-produced in several forms such as the dual-inline-pin (DIP) package. Interestingly, analog computers are now obsolete so few — if any — of the millions of op-amps that are manufactured each year spend their days calculating logarithms. Instead, most op-amps find their way into analog signal processing applications where they amplify, filter, or otherwise process many different types of waveforms.

Open Loop Op-Amp Characteristics

Figure 1 shows how to determine through simulation the open loop gain (ratio of the output voltage to the input voltage without any feedback) and bandwidth, while Figure 2 shows the simulation results. As they show, when used without feedback, an op-amp is not a particularly useful device — having a gain of 200,000 and a bandwidth of only a few Hz does not allow for very many applications.

Figure 2 also shows the midband gain[1] and the cutoff frequency[2] of the circuit; they allows us to determine an important figure of merit of amplifier performance called gain-bandwidth product (GBW). GBW is the product of the midband gain of the amplifier multiplied by the frequency at which the output voltage drops to 70.7% (or power drops by 3 dB) of the midband gain. The uA741 shown in this example has a gain of 200,000 and a bandwidth of about 5 Hz — giving it a GBW of 1 MHz. Table 1 compares this value for several low cost op-amps.

What makes the operational amplifier flexible are the feedback circuits that allow trading back and forth between gain and bandwidth. There are two basic op-amp feedback configurations that offer this: the inverting amplifier and the non-inverting amplifier. As the two



■ FIGURE 2. Simulated open loop response of the uA741 operational amplifier between 1 and 10 Hz showing the midband gain (A_{MB}) and the bandwidth and cutoff frequency (f_0).

names imply, both configurations amplify a signal, but the inverting amplifier inverts signal polarity in the process. In both cases, the overall gain and bandwidth attainable by the amplifying stage is limited by the GBW of the op-amp, as we will see in the next section.

Non-Inverting Amplifiers with Op-Amps

The non-inverting amplifier, as the name implies, does not invert the signal polarity. This amplifier is capable of producing voltage gains between unity and the midband gain value as determined by the feedback network. In this set-up, the midband gain is defined as one plus the ratio of the two resistors R_f and R_2 (Figure 3). Additionally, the input impedance in this configuration is essentially infinite (like an open circuit) which minimizes loading on the source. The milliampere output current limitation of the op-amp implies resistors in the feedback circuit are normally in the kilohm range.

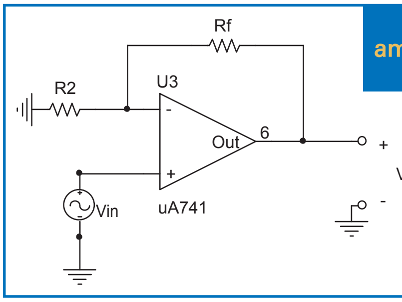
Figure 4 shows the effect of feedback on the bandwidth and gain of the non-inverting amplifier. As R_f changes from zero (short circuit) to infinity (open circuit), the resistance ratio $R_f/R_2 + 1$ changes from one to infinity. Notice that as R_f changes from zero to infinity, the feedback increases from total feedback to no feedback, gain and bandwidth are traded, with the GBW remaining constant for each combination. Also distinguishable is the GBW limitation that the op-amp itself places on the circuit independent of feedback because each of the curves must converge on the open loop curve as frequency increases.

Manufacturer	Model	GBW (typ)
National Semiconductor	LF 411	4 MHz
Texas Instruments	LM 324	1 MHz
SSG Thomson	uA 741	1 MHz

■ TABLE 1. GBW product for some low-cost op-amps.

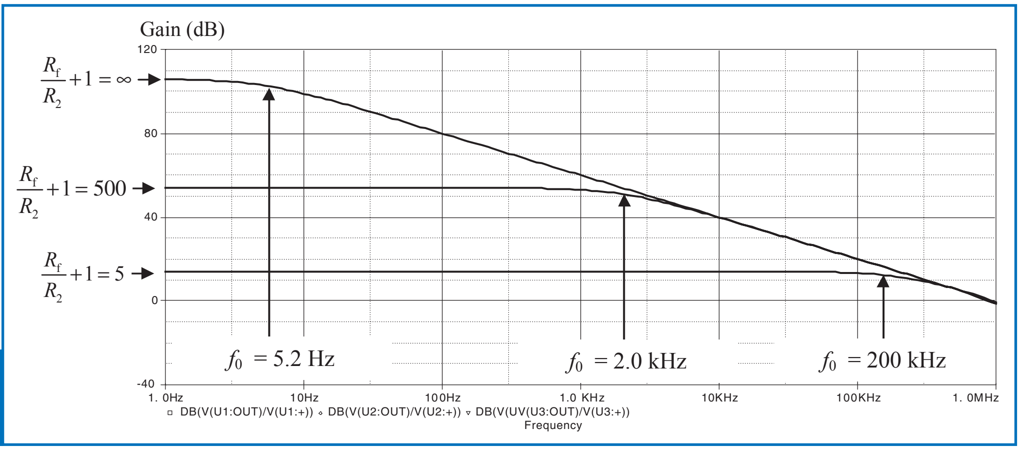
Inverting Amplifiers with Op-Amps

The inverting amplifier (Figure 5) derives its name from the fact that the amplifier inverts the signal polarity. The midband voltage gain is determined by the ratio of the feedback resistor (R_f) and the input resistance (R_1). As seen in the figure, the resistance ratio is multiplied by -1 to produce a net negative — thus causing the inverted polarity. The ratio of R_f to R_1 determines how op-amp gain and bandwidth are traded (Figure 6). If you look carefully at Figure 6, you may notice that the inverting amplifier has lower bandwidth at low voltage gain when compared to the non-inverting amplifier. This is because the inverting amplifier no longer directly trades gain for bandwidth when the amplifier voltage gain falls below about 10. This effect is most dramatic in the case of a unity gain (1.0) amplifier because the bandwidth of the inverting amplifier is reduced to half that of the

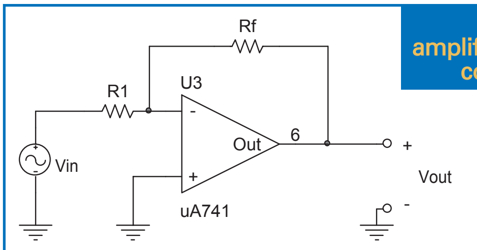


$$A_{MB} = \frac{v_{out}}{v_{in}} = 1 + \frac{R_f}{R_2}$$

■ FIGURE 3. Non-inverting amplifier configuration (power connections not shown).



■ FIGURE 4. Effect of feedback on the bandwidth and gain of the non-inverting amplifier.



$$A_{MB} = \frac{v_{out}}{v_{in}} = -\frac{R_f}{R_1}$$

■ FIGURE 5. Inverting amplifier configuration (power connections not shown).

non-inverting amplifier (not shown). Thus, the non-inverting amplifier provides twice the bandwidth when used as a unity gain voltage follower or buffer compared with a buffer constructed using an inverting amplifier.

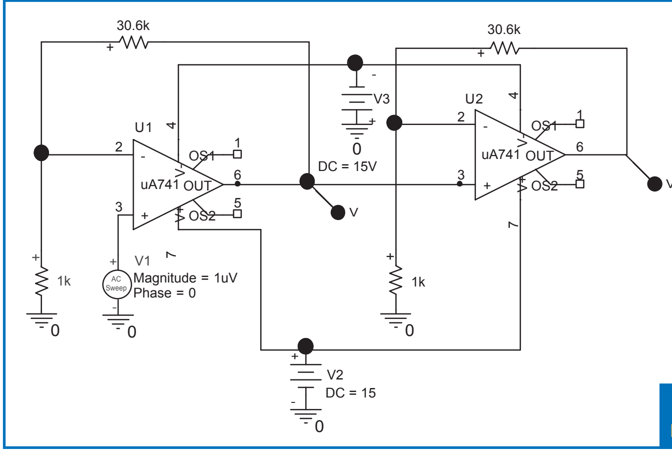
Cascading Amplifiers

Now that we have a basic understanding of how feedback circuits allow us to trade op-amp gain and bandwidth, let's apply this knowledge by considering the following scenario: If I need an audio amplifier with a midband gain of 1,000 V/V and a 3 dB or half-power bandwidth of at least 20 kHz, how do I build it? The GBW required is 20 MHz and I simply can't achieve that with a single amplifier. The solution is to cascade two or more amplifiers and divide up the gain requirement between them. Either inverting or non-inverting amplifiers can be placed in cascade (series) with each other to provide increased gain compared to a single amplifier. The very high input impedance and very low output impedance of the op-amp in the non-inverting configuration allow us to ignore the loading effect of the second amplifier on the first in the cascade. Essentially, this allows us to design a single amplifier stage and then iterate the same amplifier multiple times to attain the desired result. Figure 7 shows how a cascade of non-inverting amplifiers solves a 20 MHz GBW dilemma. For this amplifier, the overall gain is merely the product of all of the individual stage gains. You can see that the ratio of the two feedback resistors for each stage is set so that the gain is equal to the square root of 1,000 minus 1 or 30.6. The resulting bandwidth is found by dividing the 1 MHz GBW of the uA741 by 31.6, resulting in a bandwidth of 31.6 kHz — which seems more than sufficient to meet the requirement.

Unfortunately, the bandwidth of the cascade is reduced from the bandwidth of a single amplifier. This phenomenon is known as bandwidth shrinkage and is calculated using the formula shown in Figure 7 and discretely in Table 2. The BW_{out} in the figure represents the overall bandwidth of the cascade accounting for the shrinkage — where n is equal to the number of amplifiers in the cascade and BW_0 is the bandwidth of an individual amplifier.

Table 2 lists the shrinkage for cascades of two to five amplifiers. As can be seen in the table, each additional stage significantly reduces overall BW — by the fifth stage, roughly 39% of the original bandwidth is all that remains. To offset this reduction, simply increase the bandwidth of the individual amplifiers by the reciprocal of the shrinkage factor. For the two stage cascaded example, the shrinkage factor is 0.644, so increase the bandwidth of the individual amplifiers by 1.55 times over the required bandwidth to achieve the requirement. In this case, the bandwidth of a single amplifier is just sufficient since 20 kHz times 1.55 equals 31 kHz and there is 31.6 kHz of

■ FIGURE 6. Effect of feedback on the bandwidth and gain of the inverting amplifier.



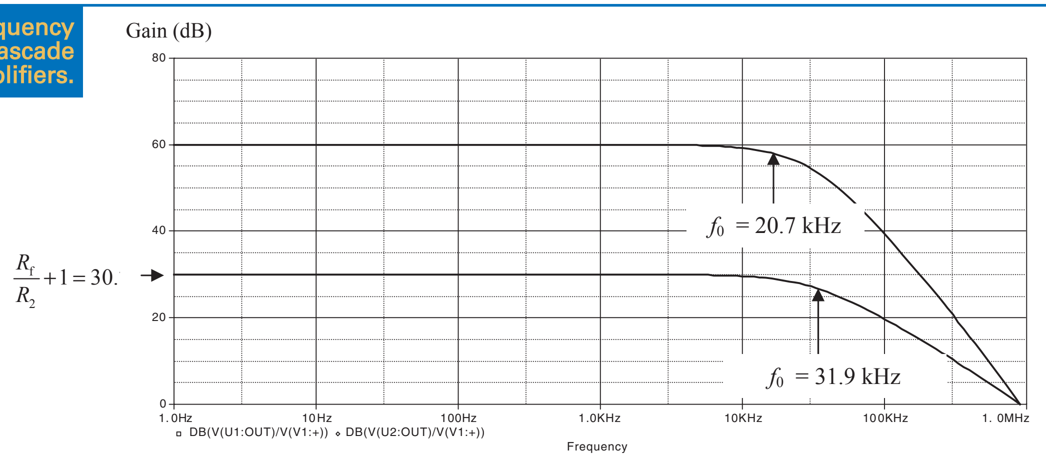
$$BW_{out} = BW_0 \sqrt{2^{\frac{1}{n}} - 1}$$

■ FIGURE 7. Cascade of non-inverting amplifiers.

Stages	BW Shrinkage Factor
1	1
2	0.64
3	0.51
4	0.43
5	0.39

■ TABLE 2. BW Shrinkage.

■ FIGURE 8. Gain and frequency response of single and cascade non-inverting amplifiers.



bandwidth to work with.

Figure 8 shows the simulation results for both the single and cascade connections of the two non-inverting amplifiers. For a single amplifier, the low frequency gain is the square root of 1,000 (30 dB) while the simulated bandwidth is 31.9 kHz — almost exactly what we predicted. For the cascade, the low frequency gain is 1,000 (60 dB) while the overall bandwidth is reduced to 20.7 kHz. This again is in excellent agreement with the value predicted by bandwidth shrinkage theory.

Example Applications

Op-amps are usually not powerful enough to take a small signal input from a microphone or head unit and amplify the source sufficiently to directly drive a speaker. For example, assume the signal is generated at a frequency of 1 kHz and the speaker requires a ±1 V peak swing at the output for adequate volume. A one volt peak swing across an eight ohm speaker (typical for a small speaker) requires that the op-amp deliver a peak current of 125 mA (1V divided by eight ohms) — which is far in excess of the uA741's maximum rating of 25 mA.

The solution to this problem is to provide additional current amplification using the complimentary Class AB emitter follower shown in Figure 9. In this circuit, resistors R3 and R4 along with diodes D1 and D2 provide base-emitter bias sufficient to set transistors Q1 and Q2 just at the edge of conduction. This is necessary in order to minimize crossover distortion as the transistors alternately switch on and off. Additionally, connecting the feedback resistor R2 across the load (R5) forces the amplifier output voltage to closely track the input signal waveform, which almost totally eliminates output stage distortion.

One can observe that the resistors R1 and R2 in the circuit set the midband gain to about 24 V/V; this should allow us to reach our necessary output swing — provided we are at a midband frequency. We can check to see if 1 kHz falls in the midband range by recalling that the uA741 has a GBW of 1 MHz. Thus, a midband gain requirement of 100 should result in a 40 kHz bandwidth. This provides sufficient bandwidth to avoid the gain roll-off that occurs as we approach the amplifier cutoff frequency.

Figure 10 displays 10 milliseconds of the output waveform that results from performing a transient analysis of the circuit when driven by a 1 kHz voltage source. The output waveform appears to be almost perfectly sinusoidal without evidence of crossover distortion at the zero crossings of the waveform. It is easy to verify this by using the Fast Fourier Transform (FFT) function of PSpice to compute the frequency spectrum of the output waveform.

In this case, the transient simulation was run for one second to create 1,000 cycles of the 1 kHz output waveform to obtain sufficient resolution in the frequency spectrum

■ FIGURE 9. Audio amplifier formed from non-inverting amplifier and complimentary transistor output stage.

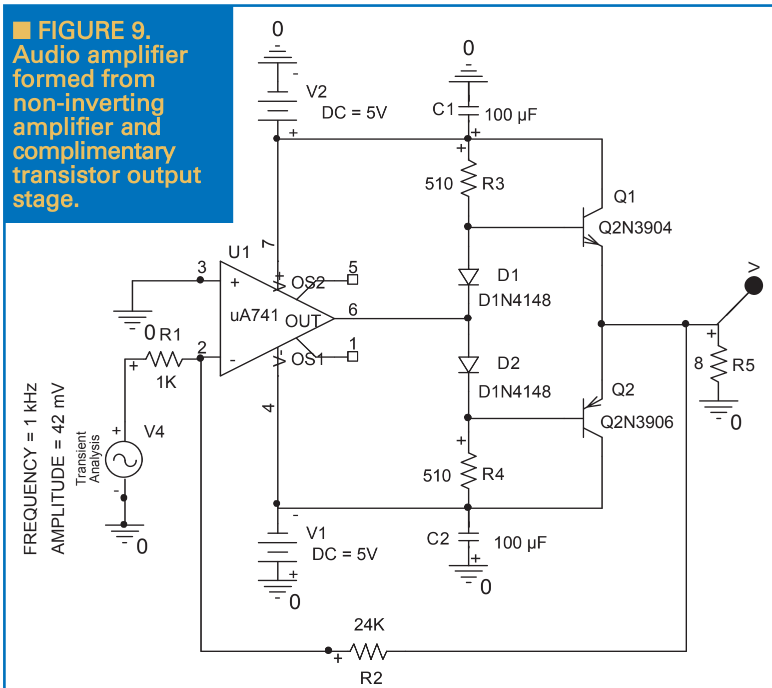


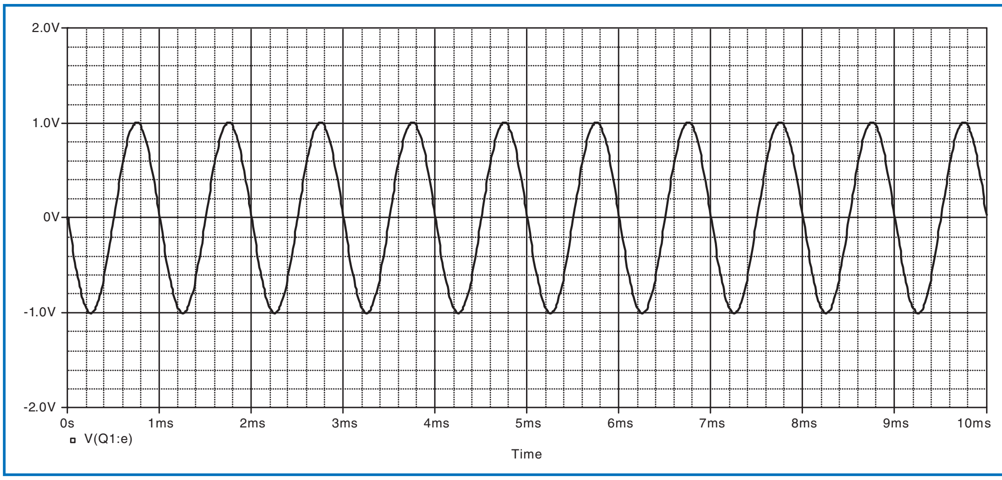
Figure 9 Parts List

RESISTORS (1/4 watt)
- R1 1KΩ
- R2 24KΩ
- R3, R4 510 Ω
- R5 8 Ω

CAPACITORS
- C1,C2 100 µF, 25V electrolytic

SEMICONDUCTORS
- U1 uA741 op-amp
- D1, D2 1N4148 diode
- Q1 2N3904
- Q2 2N3906

MISCELLANEOUS
- 5V dual power supply or batteries
- 8 Ω speaker



■ **FIGURE 10. Simulated output of audio amplifier shown in Figure 9.**

Circuit simulation software allows designers, hobbyists, and experimenters to quickly verify proper circuit operation through computer simulation before going to the time, trouble, and expense of actual circuit construction. Most software for the PC (PSpice, hSpice, etc.) is based on the program Spice (Simulation Program with Integrated Circuit Emphasis) that was developed in the late 1960s at the University of California Berkeley. Modern PC software combines the original Spice simulation code with a user-friendly, Windows-based graphical interface. This allows the circuit designer to graphically lay out the circuit by connecting the schematic symbols (schematic capture) instead of through a text file listing (net list) as the original Spice code required. A free student version of PSpice is available for download from **www.electronics-lab.com/downloads/schematic/013/**. The student version is identical to the professional version except that the number of nodes in the circuit is limited.

as shown in Figure 11. The 1 kHz fundamental has an amplitude of 1V and is clearly visible on the left side of the plot. The next most prominent signals are harmonic distortion at odd multiples of the fundamental signal (3 kHz, 5 kHz, 7 kHz, etc.), but the largest of these harmonics is only about one millivolt — three orders of magnitude smaller than the fundamental — so the amplifier's harmonic distortion is indeed very low.

Conclusion

With the proper selection of resistors, either inverting or non-inverting amplifiers can be designed to trade an op-amp's gain and bandwidth.

If you like to get your hands dirty, your next step is to build the circuits shown here in order to verify the simulation results. A simple signal generator and an oscilloscope are adequate for this task. A word of caution: due to the large open loop gain in the uA741, testing the circuit in Figure 1 requires input waveforms at the microvolt level in order to prevent clipping at the output of the op-amp. Such a low level input can be difficult to achieve in an uncontrolled benchtop environment. The other circuits are suitable even for beginners. A Parts List is included for the amplifier shown in Figure 9 for your convenience. The supply voltages are limited to 5V to avoid exceeding the transistor's maximum allowable power dissipation. The purpose of the two 100 µF electrolytic capacitors is to decouple the power supply lines to reduce the potential for oscillation. You can replace the output resistor shown in the figure (R_5) with a small eight ohm speaker if you want to use the circuit as an audio amplifier.

We hope you enjoy designing, simulating, and constructing inverting and non-inverting amplifiers built with op-amps. You may contact us with any comments or questions at luhm4e4@erau.edu. **NV**

REFERENCES

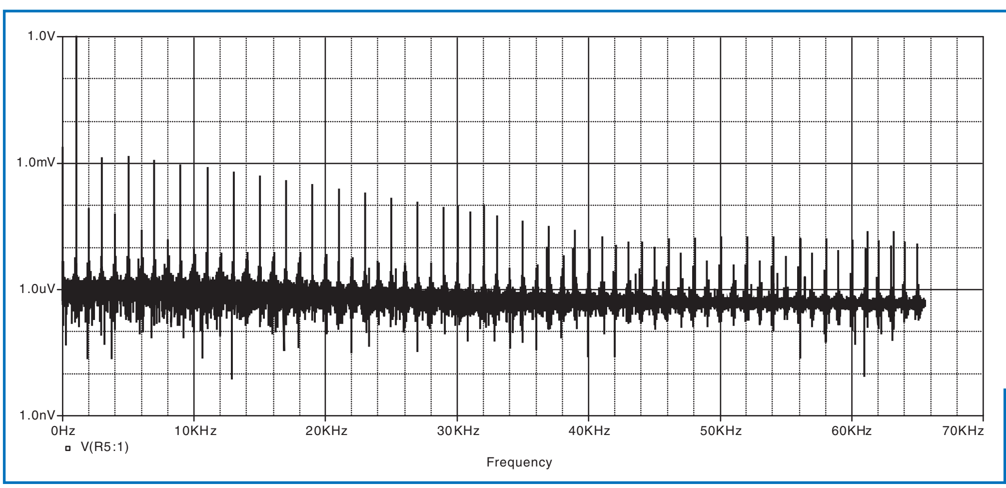
Comer, D, Comer, D. *Fundamentals of Electronic Circuit Design*. John Wiley and Sons, New York, NY, 2003.

Millman, J, Grabel, A. *Microelectronics*. McGraw-Hill Book Company, New York, NY, 1987.

FOOTNOTES

[1] Midband gain is the ratio of the output voltage to the input voltage at frequencies in between the low and high rolloff frequencies. Because an op-amp experiences no low frequency rolloff, the midband gain is the same as the gain at DC.

[2] The cutoff frequency is the frequency at which the voltage gain drops to 70.7% of the midband value. This is equivalent to a 50% drop in power gain or a reduction of 3 dB. In an op-amp — which amplifies down to DC — the cutoff frequency and the bandwidth are identical.



■ **FIGURE 11. Frequency spectrum of the output waveform of the audio amplifier shown in Figure 9.**

HOW TO DUAL BOOT Microsoft Windows XP with Vista

MOST NEW COMPUTER SYSTEMS COME WITH MICROSOFT WINDOWS VISTA PREINSTALLED. THE VISTA OPERATING SYSTEM PROVIDES MANY NEW FEATURES, BUT SUFFERS FROM THREE SHORTCOMINGS:

1 Lack of support for many devices now deemed legacy; 2 Some applications won't run on Vista; and 3 Vista requires more processing power and therefore your applications run slower.

by Michael Simpson

Vista requires that many hardware manufacturers rewrite their drivers in order for the hardware to run properly. Since rewriting drivers costs time and money, many manufacturers have opted to drop support for some of their older products.

Let's look at a real world example. I recently purchased a new machine that came with Vista installed. The system it is replacing is used to run my business. It has three printers and several applications needed to perform its daily tasks. Two of the printers were not supported nor would they ever be. Both printers were less than two years old, but if I were to continue using them in my business, I would need to purchase new versions of each.

This was just the start. Several other applications would not work and again, new versions would have to be purchased. I was going to end up spending more on replacement peripherals and software than I did on the new machine I just purchased.

So, say you're not a business and you just like to play games. Vista has been touted as being the next major leap in gaming. Well, I'm sorry, but I have not found this to be true. Most new games have a very hard time running efficiently under Vista. For example, a new game I had purchased would only run under minimum settings. The system I was using was a very fast gaming system with a high-end graphics card. On the same system running XP, I can use the highest game setting, and it runs smoothly.

While I have a love/hate relationship with Vista, I did not want to completely remove it since it is supposedly the next-generation operating system. As Microsoft releases more updates and manufacturers update their drivers, its operation will certainly become more efficient. I remember when XP was first released. It had many of the same problems as Vista does now. With the release of the third service pack for XP, it has become a very stable operating system.

So the question is, how do I add a copy of XP to my computer and still keep my Vista install intact? Normally, this can be done by installing XP then installing Vista as a second OS. However, this is not possible with an OEM install of Vista. You don't get normal Vista install disks and can only install Vista with a total reformat of your hard drive. Fortunately, there is a way to install XP alongside Vista in a dual boot configuration. In this article, I will show you step-by-step how to do just that.

Let's Begin

There are two obstacles that must be overcome when installing XP on a Vista-based system. The first is the actual install of the OS. The second is locating the XP drivers for all the hardware.

I just purchased an HP tx2000z laptop. This laptop has quite a bit of hardware, including both tablet and touch screen hardware, so locating the XP versions of the drivers will present a worst case scenario.

I suggest you purchase an external hard drive and some backup software and do a complete backup of your computer before starting the procedures I'll be covering. USB hard drives have dropped considerably in price and, if you aren't backing up your system, you are running on borrowed time.

The cheapest way to get an external backup drive is to purchase an external case. This case cost me $15 and I placed an old IDE hard drive I had inside. I have seen 160 GB hard drives on sale at my local computer store for less than $50. For a more portable solution, you can use small 2.5" enclosures.

You will also need to purchase some backup software. I have used each of the following with success:

- Norton Ghost
- Acronis True Image
- Spotmau PowerSuite

In addition to the backup software, you will need a way to make various changes to the partitions on the hard drive. For this reason, I recommend the Spotmau PowerSuite 2008.

If your computer has an option to burn recovery disks, I recommend you do that, as well. We will be performing

some risky operations so it's important that you have some sort of back up strategy in place before you start. As a minimum, back up your important documents to a Flash drive.

Obstacle 1: Installing XP

Prerequisites

Before we get started, there are a few basic tasks you will need to perform:

- Partition Utility — This utility is needed to mark your Vista boot partition hidden and inactive. This is crucial to the successful installation of XP as a second partition over Vista. The Spotmau Powersuite 2008 software can also be used to back up your original partitions.

Visit **www.spotmau.com** for pricing.

- Microsoft XP with SP2 — You will need a legal, bootable copy of Microsoft XP with SP2. If you have a copy of XP but no SP2, all is not lost. There is a program called nLite, however, which makes it easy to add SP2 or other drivers to your XP install. You will still need a legal copy of Microsoft XP and a SP2 disk, though. The program can be found at **www.nliteos.com/index.html**.
- Vista Boot Pro — Vista Boot Pro is a Vista boot manager. It makes it simple to edit or add new boot entries to Vista. This program can be found at **www.vistabootpro.org**.

In addition to the above programs, you need to test your XP boot CD/DVD to make sure your drive controller is supported by the XP installer. Some Intel-based SATA drivers are not supported. In particular, the HP dv9000t laptops will not support the base XP installer. You can do a test by booting from the XP install CD; if you make it to the partition screen, you're okay. If you get a screen that tells you there are no drives present, you will have to make some changes to your XP install.

If you do need to make changes, take a look at the nLite program I mentioned earlier. I was able to create an XP boot installer that combined my XP, SP2, and SATA drivers in order to install XP on my HP dv9000t laptop.

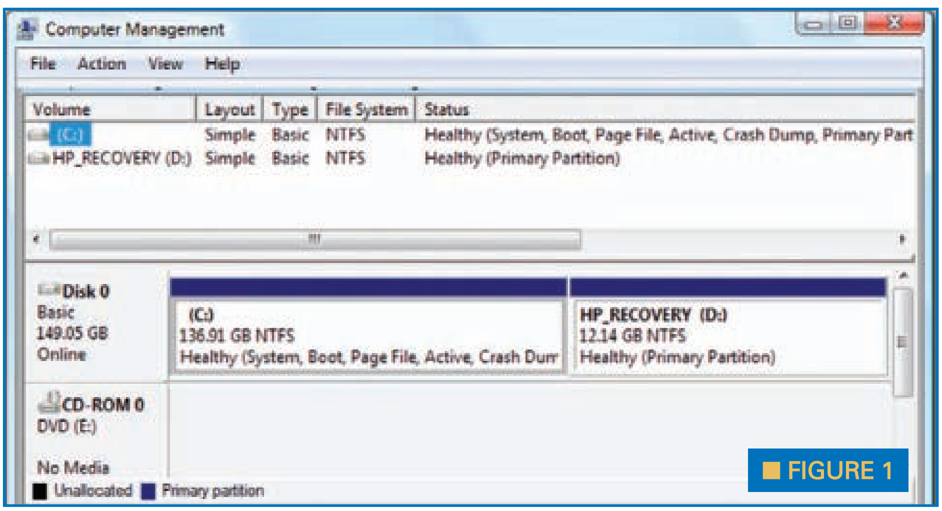


FIGURE 1

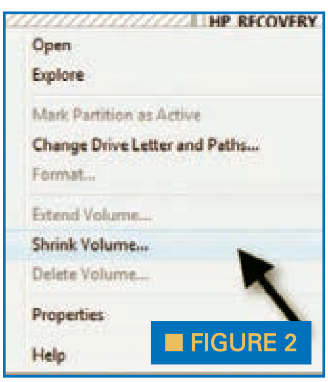


FIGURE 2

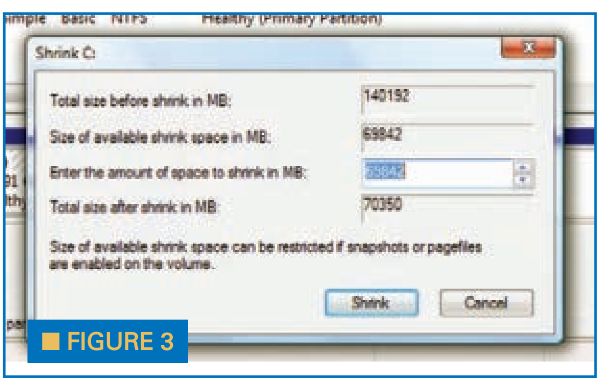


FIGURE 3

Okay, you have all the prerequisites and are ready to start the install. At this point, I will only warn you one last time ...BACK UP! BACK UP! BACK UP! With that said, let's get started.

Step 1: Reclaim Disk Space

In order to install XP, we need to make some room for it. In my case, I used the included disks and started from scratch. When you get a new PC, it has what is referred to as bloatware. Bloatware consists mostly of introductory or demo software. Some systems will include as much as 20 GB of this stuff. If you are starting from a fresh install, you will need to go through each of the applications and decide what you want to discard. Once decided, use the "Programs and Features" in the control panel to remove the applications you don't want. A good place to start is to remove expired trial software.

The next step is to clean the main drive. In most cases, this will be the C drive. From the start menu, choose the All Programs/Accessories/System Tools/Disk Cleanup option to begin this operation. Make sure you select "Files from all users on this computer" and click Continue on any security prompts, then select the drive you want to clean. After the computer scans the drive, you will be prompted for a list of selections. Here, I made sure I added the hibernation files. This almost always removes 1-2 gigabytes. Note that it will also remove the hibernate option from Vista. I will show you how to get it back later.

Step 2: Resize Vista Partition

Open up the Disk manager located at "Control Panel-Administrative Tools-Computer Management." You will see something that looks similar to Figure 1. I have removed some of the side panels to make the form easier to follow.

Our goal here is to free up enough space so that we have room to install a copy of XP. You have already removed some of the unwanted applications and cleaned the disk. This should have freed up some space so we can resize the partition. Vista has a resize feature

that will let us do just that.

Right-click on your Vista partition (in this case, C drive) and select “Shrink Volume” as shown in Figure 2. This will pop up the form shown in Figure 3. Vista is now supposed to determine the maximum it can shrink the drive and place that value in the editable field shown. You can look at the other fields to see what will be left for Vista. For XP, I recommend at least 20 GB, but it all depends on how large your drive is and what you plan on doing with XP. In my case, I am going to take the default. When satisfied, hit the Shrink button.

Once Vista has completed the change, you will have a new unallocated partition located at the end of your Vista partition as shown in Figure 4. Vista may not allow you to get the space you want. In this case, you have a couple of options. First, you can remove your recovery partition. In my case, it freed up an additional 12 GB. Vista has a page file that can keep you from resizing if it’s located near the end of the partition. If this happens, you have a couple choices. First, you can remove the pagefile.sys file then have Vista build a new one. The easier way is to back up the drive then restore it. Most backup software will not back up the pagefile.sys file. Vista just restores the file when it sees it’s missing.

My recommendation is to remove the recovery partition. Before you do so, make sure you have a backup or a set of backup disks. Removing this partition makes for a cleaner install and yields more space for your XP partition.

Step 3: Prep New Partition

Right-click on the new partition and select the “New Simple Volume” option as shown in Figure 5. Select the default options until you get to the Format Partition form shown in Figure 6. Change the volume label to XP and check the “Quick Format” option. Hit the Next button and finish the operation. After a couple of minutes, Vista will transform the partition into a new drive.

When this is all said and done, you should have two partitions on your hard drive configured as shown in Figure 7. Exit the Computer Management form and prepare to boot from the Spotmau power suite software. Before you reboot, you may want to change the drive label for the Vista partition. If you change it to “Vista,” it will make it easier to identify the partitions later.

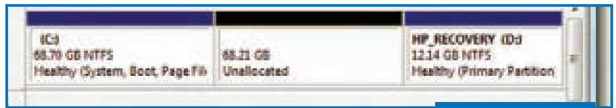


FIGURE 4

FIGURE 5

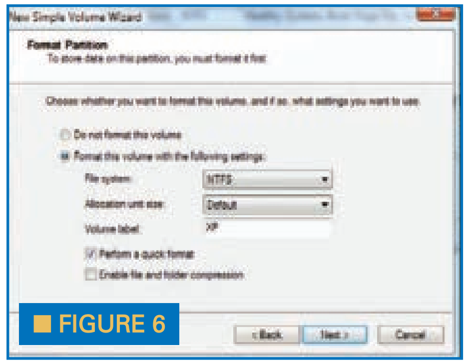


FIGURE 6

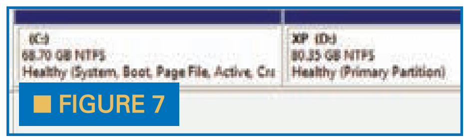


FIGURE 7

Step 4: Hide Partitions

At this point, you need to hide the Vista and Recovery partitions from the XP installer. If you don’t, they will get all mucked up. Insert your partition utility boot software and restart you computer. (Another product you can use for this task is the Acronis Disk Director Suite.) When PowerSuite 2008 starts, select the advanced mode. You will be prompted for various boot choices such as resolution and color depth. Choose whatever best suits your computer. I chose the VBE option when prompted for the display driver.

When the application screen shown in Figure 8 is presented, select the Partition Genius application. Right-click on the Vista partition and select “Hide Partition” from the Advanced menu as shown in Figure 9. It will give you various prompts; just click the Apply button as each one pops up. By hiding this first partition, we are telling the XP installer to ignore this drive when we do the install. Failure to do this will render the Vista partition unbootable and will totally muck up the XP installation, as well.

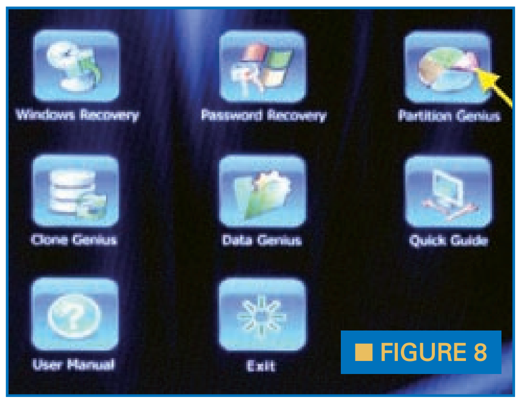


FIGURE 8

Step 5: Install XP

Exit the current form by hitting the small X in the top righthand corner. When presented with the application screen, click the Exit icon. You will be prompted to hit Enter when it’s ready to reboot. At this point, you need to remove the Spotmau disk and insert the XP Install disk.

Reboot the machine to start the installation process. It will take a few

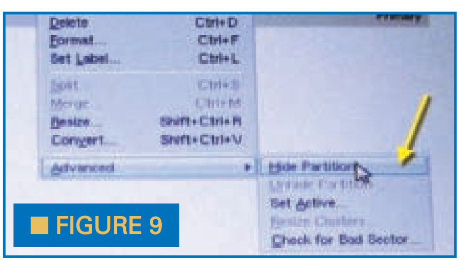


FIGURE 9

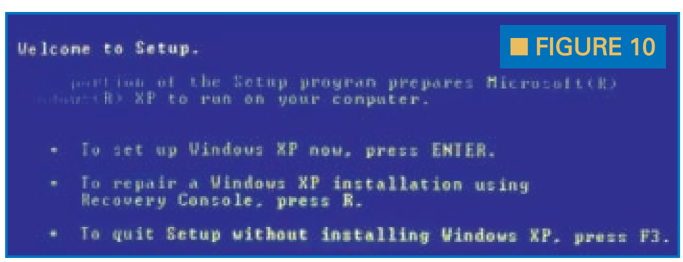


FIGURE 10

FIGURE 11

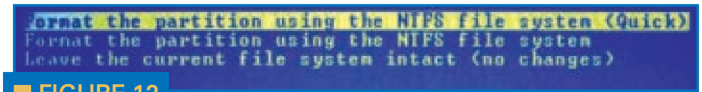


FIGURE 12

minutes for the XP installer to load, but eventually you will be presented with the Welcome to Setup form shown in Figure 10. Hit Enter to continue. Next, you have the license agreement. Read the form, then hit F8 to continue.

You will now be presented with the Partition form shown in Figure 11. Select the partition labeled XP using the cursor keys and hit Enter. If you get a warning about the active partition, just ignore it and hit Enter to continue.

You will then be prompted to format the partition as shown in Figure 12. Select the Quick Format and hit Enter. The drive will spend a few minutes formatting then start the copy and install process. On a fast computer, it will take 30-40 minutes. Near the end, you will be prompted for basic information such as Time Zones. It all depends on the exact hardware that is detected. For instance, if the installer detects your network interface it will prompt you for network information. In my case, the drivers for the network were not detected so I was not prompted. Just keep in mind the install will vary somewhat for each computer.

When the installation is finished, you will be placed on the XP desktop. You need to immediately reboot and restore the hidden Vista partition. Do this by booting up from the Spotmau Power Suite CD. Follow the same prompts as before and enter the Partition Genius application.

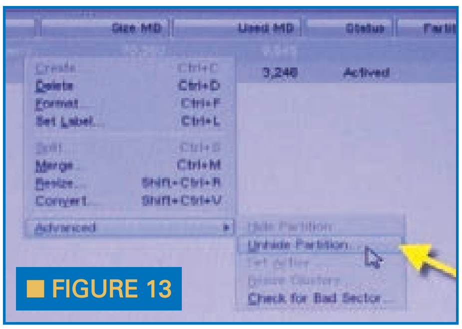


FIGURE 13

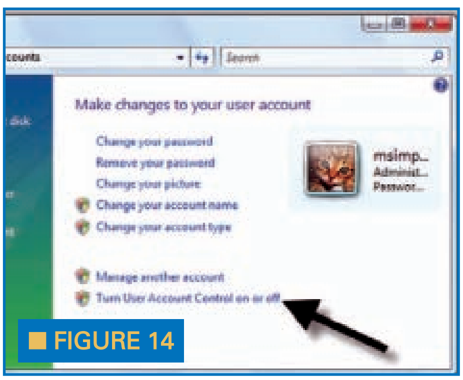


FIGURE 14

Step 6: Unhide and Activate the Vista Partition

Right-click on the Vista partition (first one) and select the Unhide Partition option shown in Figure 13. You also need to set the Vista partition to active by selecting the Set Active option on the Advanced menu. Exit the form and application menu and reboot the machine. Make sure you remove the Spotmau disk.

Step 7: Tell Vista about XP

In order to perform the following tasks, we need to turn off the UAC feature in Vista. To do this, load the User Accounts form from the control panel and click the "Turn User Account Control on or off"option as shown in Figure 14.

This will load the form shown in Figure 15. Remove the check from the checkbox by clicking it, then hit the OK button. You will be prompted to reboot the machine.

Next, we need to change a few options so we can see some system files.

Start the Computer form and select "Folder and Search Options" from the Organize Menu as shown in Figure 16.

Click the View tab and set the following options (shown in Figure 17):

- Select Show hidden files and folders.
- Uncheck Hide extensions for known file types.
- Uncheck Hide protected operation system files.

Click the OK button to make the changes.

On the Computer form, double-

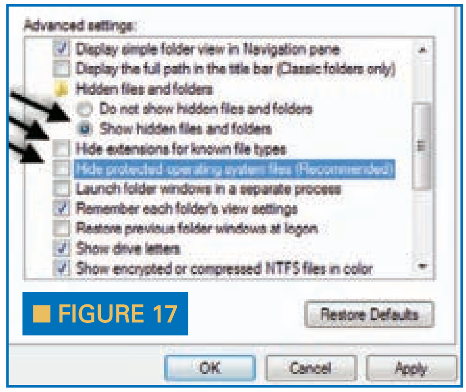


FIGURE 17

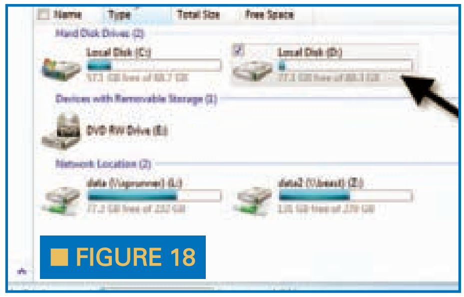


FIGURE 18

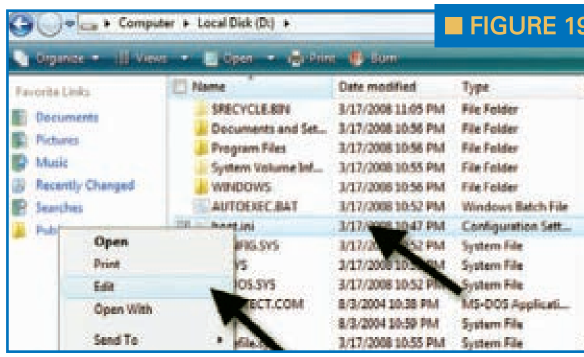


FIGURE 19

FIGURE 15

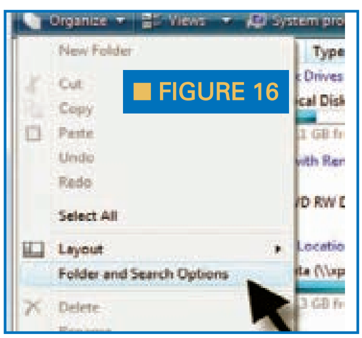


FIGURE 16

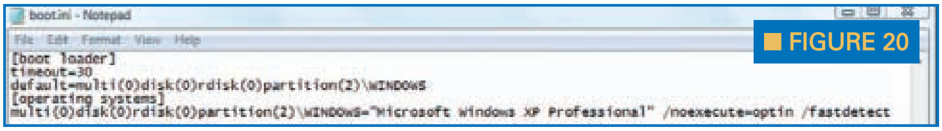


FIGURE 20

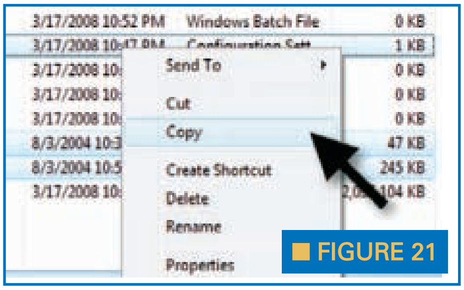


FIGURE 21

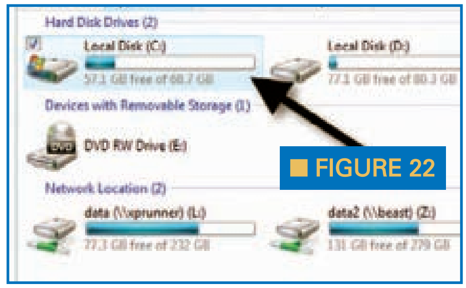


FIGURE 22

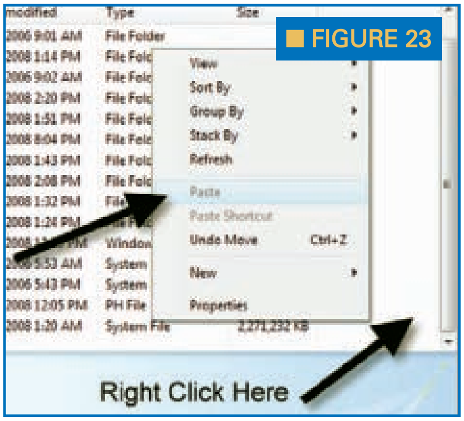


FIGURE 23

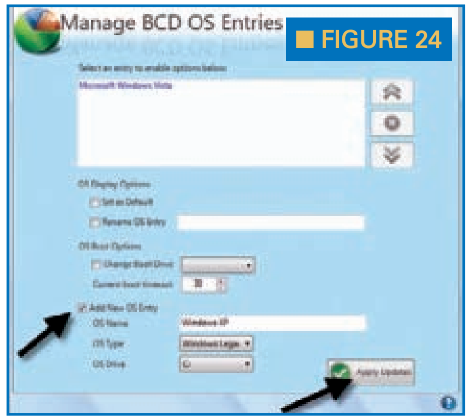


FIGURE 24

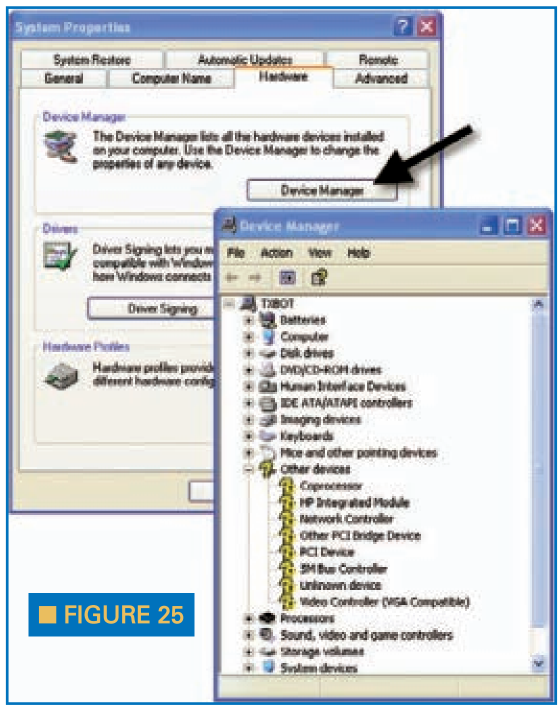


FIGURE 25

click the XP drive. This will be the second local disk, in my case, drive D as shown in Figure 18. This will open up the drive so you can see all the files and folders on it.

Right-click on the boot.ini file and select Edit from the menu as shown in Figure 19. This will open up the XP boot definition file.

Your boot.ini file should look like the one shown in Figure 20. (As a matter of fact, if you only have two partitions it should look exactly the same.) You may have to change the partion parameter if your XP is not the second partition on the drive. If you are not sure, you can open up the Computer Management-Disk Management form and look at the order in the lower portion of the pane. All partitions count and start with the number 1. If you make changes to the file, save it and close the form.

Now use the control key and select the boot.in, NTDETECT.COM, and ntdr files, then right-click. Select the Copy option as shown in Figure 21.

Go back to the Computer form and double-click on the Vista drive as shown in Figure 22.

Right-click on the form to the right of the file entries as shown in Figure 23. Select the Paste option. The three files you selected on the previous form should now show up on this form, as well.

The last thing we need to do is edit the Vista boot manager configuration file. To do this, we will use the free boot manager called VistaBootPro. Download and install it now. (The program can also be downloaded from the PROnetworks website at **www.vistabootpro.org**)

Once installed, start the program. The program will prompt you the first time it is loaded to make a backup of the Vista boot config settings. You can ignore that for now. Select the "Manage BCD OS Entries" tab. Check the Add New Entry box and fill out the three parameters with the ones shown in Figure 24, then hit the Apply Updates button.

You can now shut down the program and reboot the machine. You will be prompted at startup for the Vista or XP operating system with Vista being the default. Go ahead and reboot, and make sure you can boot XP.

That's it! You now have both operating systems installed. Your XP boot is probably not operating at 100% because you have not installed any drivers.

Obstacle 2: Updating XP Drivers

The advantage of a dual boot system is that you can use your Vista boot to surf the web and locate drivers until you get your XP network drivers up and running. However, before you start installing anything on your brand new XP drive, you should do a backup. This means either installing backup software on the drive or using the Partition to Image option on the Spotmau. At this point, you can also install the software directly to the OS so you can schedule backups.

Right-click on the My Computer entry in the Start Menu and select the Properties option. This will load the System Properties form shown in Figure 25. Select the Hardware tab and hit the Device Manager button and the Device Manager form will load. It is this form where you can check the

status of your drivers. My tx2000z HP latop is missing eight.

Where Do I Start?

The first place to look is your system's manufacturer's website. In my case, this is HP, so I went to www.hp.com and typed "tx2000z drivers" in the search field on the main page. Unfortunately, only a couple drivers were located there. After doing some research, I know that the ts2000z is an upgrade from the older tx1000 laptop. Many of the hardware items on the website are the same. A search for "tx1000 drivers" yielded a few more drivers, including the Broadcom wireless network driver.

I also found out that my laptop motherboard uses an Nvidia Chip set. Nvidia has an option where they can detect your hardware. This enabled

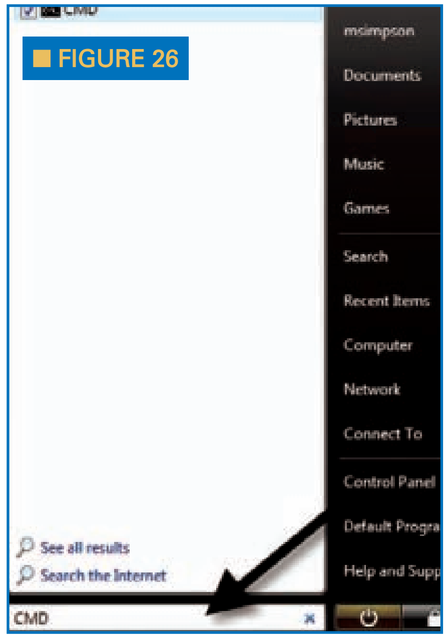


FIGURE 26

me to download and install the motherboard chipset drivers. This included the wired network driver. I also downloaded the Nvidia 6150 display driver. I found that this driver must be installed by manually selecting the Display Driver from the list and clicking the Have Disk button. I then pointed to the location where I unloaded the display drivers I downloaded from Nvida.

If your PC is using an AMD processor, you will need to download the AMDCPU driver that is located on the AMD website (www.amd.com/us-en/). Make sure you get the 32-bit or 64-bit version to match the OS you are running. For normal XP, this will be the 32-bit version. This driver will allow various utilities to control the processor speed. This is a must for a laptop. It also allows the processor to enter.

Now, unless you are installing XP on a tx2000z laptop you are probably going to have different driver issues. You have to do a little research. Look for other models that are similar to your own. In many cases, a trip to the manufacturer's website of the device you are trying to find a driver for will be all that is needed.

By all means, save and document all your drivers. As I mentioned at the beginning, a Flash drive makes a good temporary storage medium. If you place the drivers on your hard drive,

don't forget to back them up in case you decide to restore from one of your backups.

Now for the clincher. Many times, the order in which the drivers are installed is important. This is why I told you so many times to do a backup. This gives you a point to fall back on if you can't get a particular driver to work. As a rule of thumb, I install the chipset drivers first, then the display drivers. Next, I install the wireless so I can get the network up and running. Sound drivers are next, then I just start working through the rest of the bunch.

Also, if you get a device that does not work, go back and take a closer look at the System Devices form. Often, you will find a driver that has to be reloaded manually. This happens frequently with the high definition audio drivers.

Final Thoughts

Hopefully, this information will get you well on your way to creating a dual boot configuration on your current Vista machine. Let me close with a few comments.

Turn Vista Hibernation On

If you removed your hibernation file back when we were trying to gain space, you may want to turn the hibernation feature back on. First, you need to open up a command window by typing CMD in the search field at the bottom of the Start Menu as shown in Figure 26. Once your command window is open, type in the following command at the prompt and hit enter:

```
Powercfg /hibernate on
```

You need to do this with the UAC turned off.

Change Default Boot OS

Use Vista Boot Pro to change the default operating system or the time-out period. It's pretty simple.

In the Manage BCD OS Entries tab, click on the OS you want to become the default and select the Default checkbox and click the Apply button. You can also change the order that the OS appear by using up and down arrow buttons on the side of the form.

Reset Hidden Files

You may want to reset your hidden file types so that certain files are not displayed, as these can show up on your desktop. Just reverse the process I showed you back in Step 5.

Last Words

Before I end this article, I must ask that you don't contact me with questions on Vista or XP. I am by no means a Vista or XP guru. The things I showed you here I found by trial and error and by searching the web. If you run into problems, restore your backup and try again. (If you did not create a backup, all I can say is, I TOLD YOU SO!) Hopefully, you will be as successful and satisfied as I have been with a dual boot system. **NV**

SMILEY'S WORKSHOP

AVR MICROCONTROLLER: C PROGRAMMING - HARDWARE - PROJECTS

by Joe Pardue

Get started now with this series!
Joe's book & kit are available in our webstore at **www.nutsvolts.com**

PART 3:
C Types, Operators, and Expressions

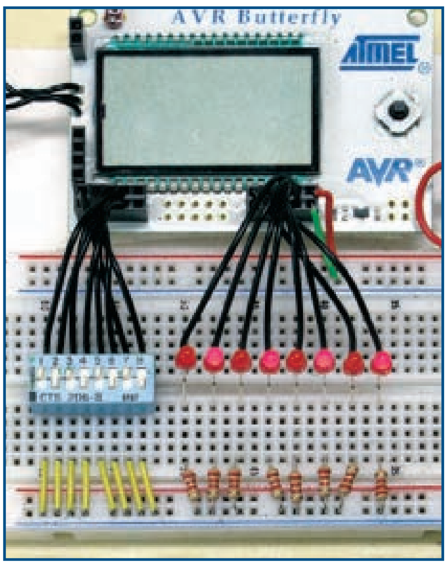
Last month, we wrote our first C Program: CylonEyes.c, built the CylonEyes hardware, and watched the LED light zip back and forth.

This month, we will look at AVR port input and output. And we will add an eight-bit DIP switch to our learning platform. Remember in Workshop 1 where I compared learning C and AVRs to walking across two continents and said I'd give you some warnings? Well, prepare to have a branch smack you in the face. This workshop has some very tedious facts in it. Sorry, but it can't all be fun and games and you have to climb a high wall in order to get to the fun stuff. As Randy Pausch said in the 'Last Lecture' "... brick walls are there for a reason. The brick walls are not there to keep us out. The brick walls are there to give us a chance to show how badly we want something. Because the brick walls are there to stop the people who don't want it badly enough. They're there to stop the other people."

Data Types and Sizes

Bits — The first computers were people with quill pens who spent their lives calculating tables of things like cannonball trajectories to help soldiers more accurately slaughter their enemies. Later, mechanical computers with brass gears and cams were developed to make the slaughter cheaper, quicker, and easier. Then one day a genius figured that you could do all this computing even easier if you used switches.

Switches can be off or on, and the fundamental datum is the 'bit' with exactly two 'binary' states. We variously refer to these states as '0 and 1' or 'on and off' or 'clear and set' or 'true and false.' It's the latter that allows us to use bits to automate Boolean Logic (an 'algebra' for determining if a statement is true or false), and thus the modern binary logic computer entered the world and now slaughter is so cheap, quick, and easy to compute that anybody can do it. Maybe this is skimming the topic a bit (har!) but a full explanation would begin with the first sentence of Genesis and only hit its stride about the time Alan Turing offed himself as his unjust reward for saving the free world in WWII. And while fascinating, it won't get us blinking LEDs any quicker, so let's move on.



■ FIGURE 1. Port input and output.

Each of our LEDs is connected to a microcontroller pin that can have two voltage states: ground or Vcc, which can be manipulated as a data bit.

Bytes — The AVR and many other microcontrollers physically handle data in eight-bit units called bytes — a data type that can have 256 states, 0 through 255. This is shown in the following sequence of states (leaving out 5 through 250):

00000000 = 0
00000001 = 1
00000010 = 2
00000011 = 3
00000100 = 4
...(states 5 through 250) ...
11111011 = 251
11111100 = 252
11111101 = 253
11111110 = 254
11111111 = 255

For CylonEyes, what you are seeing is eight of the 256 possible states being presented in a sequence that fools us into thinking we are seeing a back and forth scrolling motion. Using binary numbers where the lit LED is represented by 1 shown next to the binary, hexadecimal, and decimal equivalent, what we are seeing is:

00000001 = 0x01 = 1
00000010 = 0x02 = 2
00000100 = 0x04 = 4
00001000 = 0x08 = 8
00010000 = 0x10 = 16
00100000 = 0x20 = 32
01000000 = 0x40 = 64
10000000 = 0x80 = 128

In microcontroller applications, we will often be dealing with the states of byte-sized ports, like port D. A microcontroller port is a place where outside voltages (0V or 3V) can be read or set. Experienced microcontroller programmers memorize the binary equivalent of hex digits and find hex numbers very useful. For instance, given 0xA9, what would the LEDs (or the voltage states of an eight-bit register) look like? If you memorize the hex table, you come up with 0xA = 1010 and 0x9 = 1001, so the LEDs (voltage states) will look like 10101001. If you ask the same

question in decimal, what will 169 look like on the LEDs? Well, good luck on doing that in your head.

char **—** The name of this data type is short for character, and is typically used to represent a character in the ASCII character set. Originally, there were 127 ASCII characters used by Teletype machines to transmit and receive data. Remember that in Figure 1 of Workshop 1, you saw Dennis Ritchie who wrote C standing next to Ken Thompson who wrote UNIX, working on a Teletype machine. Clunky as they were (the Teletype, not Ritchie and Thompson), Teletypes were light-years ahead of entering data by individual switches representing each bit of data. Teletypes send and receive characters so a lot of C, especially the standard library, stdio, is character oriented. The number of bits in a char is machine dependent, but in all machines I've encountered including the AVR, a char is an eight-bit byte that can have 256 bit states. The computer uses this byte of data as representing a signed value from -128 to +127.

unsigned **—** If the modifier unsigned is used in the definition of a char variable 'unsigned char,' the value is from 0 to 255. Many C compilers will have 'byte' or 'Byte' defined as equaling unsigned char. The byte keyword is not part of C, but it is very convenient, since in microcontrollers we usually use a lot of numbers, but not a lot of 'char'acters.

int **—** On AVR microcontrollers, int (short for integer) declares a 16-bit data variable as having values from -32768 to +32767. A variable declared with 'unsigned int' will have a value from 0 to 65535.

The long and short of it **—** Everybody else makes that dumb joke at this point, so why be different? You can declare variables as 'short int' and 'long int.' For C, the size is machine dependent, but on many systems a short int is the same as an int (16 bits), while a long int is 32 bits.

uint8_t and uint16_t **—** More recent C standards use uint8_t for an eight-bit unsigned char and uint16_t for unsigned int. This increases portability between machines and lessens confusion about the size of the data type we are using.

Declarations

A declaration is a text statement that declares to the compiler how your words are to be used. When you declare 'unsigned char counter = 0' you are telling the compiler that when it encounters the word 'counter' to consider it as data stored at some specific address with the alias name 'counter' that can have values from 0-255, but in this case, it initially has a value of 0.

Operator	Name	Example	Defined
[]	Array element	x[6]	Seventh element of array
.	Member selection	PORTD.2	Bit 2 of port D
->	Member selection	pStruct->x	Member x of the structure pointed to by pStruct
*	Indirection	*p	Contents of memory located at address p
&	Address of	&x	Address of the variable x

TABLE 1. Data access and size operators.

Operator	Name	Example	Defined
()	Function	wait(10)	Call wait with an argument of 10
(type)	Type cast	(double)x	x converted to a double
?:	Conditional	x?y:z	If x is not 0 evaluate y, otherwise evaluate z
,	Sequential evalution	x++,y++	Increment x first, then increment y

TABLE 2. Miscellaneous operators.

Variable Names

The changeable data you are processing is stored in bytes of RAM (Random Access Memory) at specific addresses. Variables are names that provide an alias for the address being used. We'll look at those gory details in a later Workshop.

Constant Names

Literal constants are data that are not really variables but read-only values that are textually replaced on the first pass of compilation. The literal's name is traditionally typed in all caps, and located in a header file or at the start of the software module. For example, we might want to use pi in a calculation, so we define as follows:

```
#define PI 3.1415926
```

We can then use PI anywhere in our software and the compiler will automatically substitute the numerical value for it (Listing 1).

Operators

Note: Some of these operators may seem strange at this point, but they are explained fully in later Workshops. Then they'll seem even stranger. You may already be familiar with some of these operators, and rather than take space elaborating each, we will wait to see how they are used in context in later Workshops.

We will save the Arithmetic operators for Workshop 4 and in Workshop 5 we will look at Logical, Relational, and Bitwise operators.

Okay, I'm tired of all these words, let's play with some hardware!

Port Input and Output

We skimmed over a lot in Workshops 1 and 2 so that we could get some LEDs blinking. Let's now take a more detailed look at I/O ports.

AVRs are available with from six to 86 I/O pins (ATmega169 has 54 I/O pins). Most of these pins are organized into eight-bit ports. Many of the pins have more than one possible function. They can be used to input or output digital logic data, they might be used for detecting external interrupts, or for analog-to-digital conversions and so on.

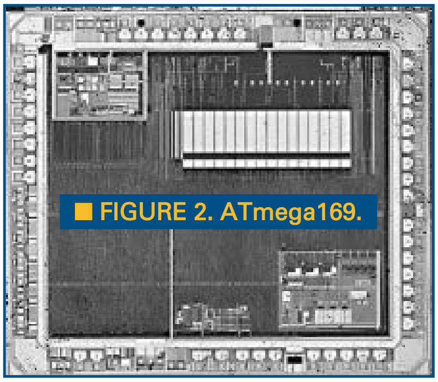
Pins can be set as either Input or Output. Pins set as outputs can be driven to Vcc or Gnd. Pins set as inputs

```
float pieCircumference = 0.0;   // we don't know yet
float piePanRadius = 2.2;       // this we measure so it can vary

// sometimes you just gotta have PI.
pieCircumference = PI * ( piePanRadius * 2 );
```

Listing 1

Listing 2

```
// PortIO.c
#include <avr/io.h>

int main (void)
{
        // Set port data direction registers
        DDRB = 0x00;  // PORTB set to input
        DDRD = 0xFF;  // PORTD set to output

        PORTB = 0xFF; // enable pull up on input port

        while(1)
        {
              PORTD = PINB;
        }
}
```

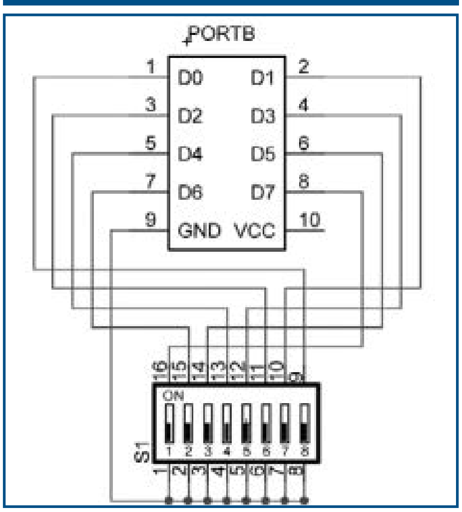


■ FIGURE 2. ATmega169.

can have a pull-up resistor enabled or disabled. Each pin can be changed to either input or output without affecting the state of any other pin. This is also true for changing an output pin's drive value or an input pin's pull-up resistor enable or disable.

The ATMEGA169 on the Butterfly has six eight-bit and one four-bit general-purpose I/O ports as shown in Figure 2 (page 3, ATMega169 data book). (You can get the data book from **www.atmel.com** or as part of the Workshop3.zip*). That diagram looks mighty complex, doesn't it? Well, it is a simplified block diagram of a circuit that is vastly more complex. When you see a photomicrograph of these chips like in Figure 2, they resemble aerial photos of a vast ancient city with streets laid out in a grid surrounded by a wall. The ports are like the gates to the city. Each pin in a port is configured by three individual bits in registers DDRx, PORTx, and PINxn (see page 74 of the datasheet). For example, port A has PORTA, DDRA, and PINA registers each with eight bits to set up each pin. When used for general-purpose I/O, the port DDRx (Data Direction Register) bits must be set to 1 for output or 0 for input. For example, to use the upper four bits of PORTB as inputs and the lower four bits as outputs, set the bits to 00001111, which in hex is 0x0F:

■ FIGURE 3. PORTB to eight bit DIP switch.




```
DDRB = 0x0F;
```

When a port pin is configured as an input, writing a 1 causes the port pin to be connected to Vcc via a resistor. If it is configured as an output pin, then writing 1 drives the port high (logic 1) or writing 0 drives it low (logic 0).

```
// enable pull up
// on input port
PORTB = 0xFF;
```

If you look at the I/O-Ports section of the ATmega169 datasheet beginning on page 52, you will see that there is a bit more complexity to this topic than we covered here, but at least we have enough for now to read and write ports. One thing to remember is that we set a port to a value by writing to the PORTx, but we read a port by reading PINx. For example:

```
PORTD = PINB;
```

writes the value off the PORTB pins to PORTD. It is very common for newbies to forget this. In this project, we will set PORTB to input data from switches and PORTD to output Vcc to drive LEDs. We use the PINB register to read the switches from port B and write the value to port D using the PORTD register. Then we write an infinite loop that gets the switch data from port B using PINB and equates it to PORTD that will light the LEDs.

Create a new AVRStudio project — PortIO.c — as discussed in Workshop 2 and enter the C code shown in Listing 2. The PortIO source code is also included in the Workshop3.zip.

And as before, compile and download the hex file to the Butterfly. Remember to turn the Butterfly off and back on WITH THE JOYSTICK PRESSED TO THE CENTER! Then, hold down the center joystick button while clicking on the 'AVR Prog...' menu item in AVRStudio. Finally, after the code downloads, remember to turn the Butterfly off and back on, then click the joystick to the upper position to start the program. DOWNLOAD THIS PROGRAM TO THE BUTTERFLY BEFORE WIRING PORTB, WHICH INTERFERES WITH THE JOYSTICK.

Using the LED wiring from the last project, add the switch as in Figures 1 and 3. You may refer to Figure 3 from Workshop 1 to see where port B is (you can download Workshop 1 from **www.nutsvolts.com** or **www.smileymicros.com**).

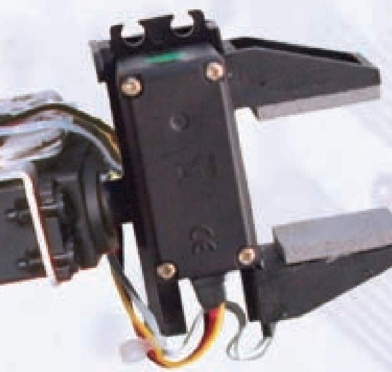
The LEDs will display the state of the switches. And, if you are paying attention, you'll say, 'Hey wait a minute, these switches and lights are reversed.' And you'll be wrong. PORTB pins are pulled up so when nothing is on them they are at Vcc and the associated LED is lit. Turning the switch 'on' pulls the pin to GND, thus the associated LED turns off. Okay, you think it's crazy that one kind of on means off, but as we'll see in detail later, we define what voltages are true or false or on or off any way we want to. A lit LED can mean either off or on, it's a choice not a law. Think about the logic behind this since it isn't nearly the most confusing thing we'll eventually see. Next month, we'll look at some more C syntax and learn how to get a Butterfly to talk to a PC. **NV**

Joe Pardue can be reached at
www.smileymicros.com

*Workshop3.zip is available from *Nuts & Volts* (**www.nutsvolts.com**) and **www.smileymicros.com**.

ELECTRONET

For the ElectroNet online, go to **www.nutsvolts.com** and click **Electro-Net**



ROBO DEVELOPMENT
CONFERENCE & EXPO '08

NOV. 18-19, 2008

SANTA CLARA CONVENTION CENTER

SANTA CLARA, CA

Join the International Technical Design and Development Event for the Personal, Service, & Mobile Robotics Industry

EXCLUSIVE OFFER:

USE PRIORITY CODE RDNAV AND

SAVE $300 *ON A CONFERENCE PASS*

WWW.ROBODEVELOPMENT.COM ■ 800-305-0634

- ***The industry's most comprehensive conference program covering these critical topics:***
  - Systems & Systems Engineering
  - Tools & Platforms
  - Enabling Technology
  - Achieving Autonomy
  - Design & Development
- ***Valuable networking opportunities that put you in touch with peers, industry experts and up-and-coming talent:***
  - Evening Welcome Reception
  - Speaker Meet & Greet
  - Birds-of-a-Feather Discussions
- ***Exposition floor featuring what's new and what's next in robotics design and development***
- ***Learn from exclusive keynote presentations delivered by world-renowned robotics industry experts:***

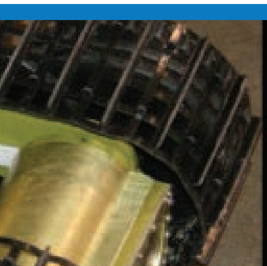
Sebastian Thrun, Winner of the DARPA Grand Challenge; Director, Artificial Intelligence Laboratory, Stanford University

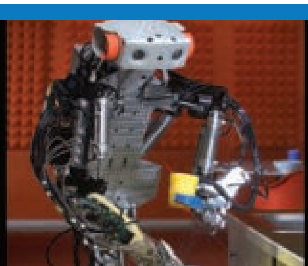
Maja J. Mataric, Founding Director, USC Center for Robotics and Embedded Systems; Director, USC Robotics Research Lab

Jeanne Dietsch, CEO, MobileRobots Inc

Michael Bruch, Section Head, Space and Naval Warfare (SPAWAR) Systems Center

FOR COMPLETE EVENT DETAILS
VISIT WWW.ROBODEVELOPMENT.COM





*For Information on **Sponsorship and Exhibiting Opportunities**, contact Ellen Cotton at ecotton@ehpub.com or 508-663-1500 x240*

JOIN THESE LEADING COMPANIES AT 2008 ROBODEVELOPMENT

Founding Sponsor

Microsoft

Gold Sponsor

MOBILEROBOTS INC

Silver Sponsors

AD Robotics

Advanced Motion Controls

All Motion

Boston Engineering

Elmo Motion Control

Energid

pico Technology

Velodyne

Willow Garage

Corporate Sponsors

FESTO

PNI

HOKUYO

Sentek solutions

Jetta Company Limited

OceanServer Technology, Inc

Unigen

Analyst, Association & Academic Co-Sponsors

Carnegie Mellon Silicon Valley

OMG Object Management Group

SCIVESTOR

VC ~ Private Equity Roundtable Meet. Connect. Invest.

Media Co-Sponsors

Circuit Cellar

Entertainment Engineering

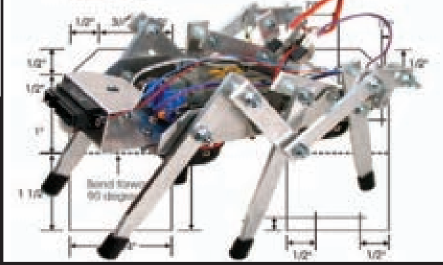
Dr. Dobb's Journal

Linux Journal

Nuts and Volts

Servo

Listing as of August 30, 2008. For a current list of participating companies, please visit *www.robodevelopment.com.*



■ BY VERN GRANER

THE PEANUT BUTTER MONSTER DETECTOR

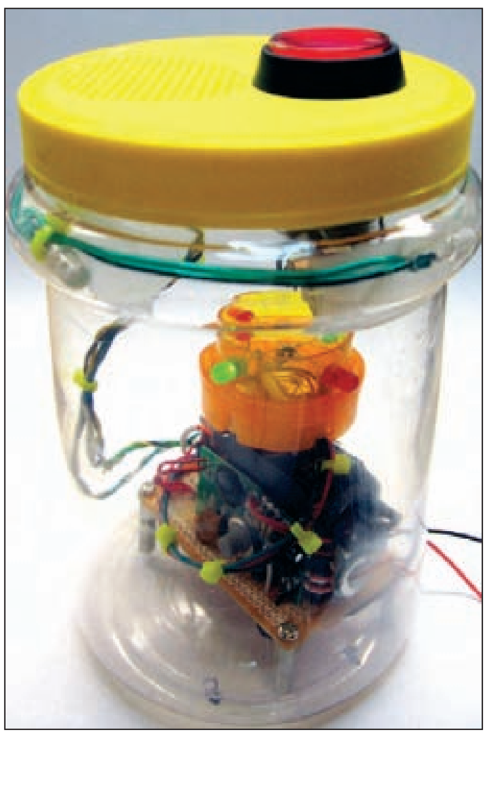
Build this robotic bedside companion to help ease your child's nighttime fears

THE PEANUT BUTTER MONSTER DETECTOR is a combination bedside night light and audible/visual effects generator for reassuring frightened children. If a child is unexpectedly awakened at night, they can easily trigger the unit with a single pushbutton. Once triggered, the unit floods the room with bright white light while it emits sounds, flashes LEDs, and rotates a scan head to the accompaniment of a reassuring spoken dialog. After the unit gives an "all clear," it becomes a bedside night-light that slowly fades out after a few minutes.

With Halloween upon us, many times kids are begging to see a scary movie or visit a haunted house, only to find out too late that it was a bit much for them. What was a fun thrill for younger kids (with friends and family around) can become a scary memory for them at bedtime. This usually manifests itself in requests that lights be left on or doors be left open. When the sun goes down and the scary fun becomes just plain scary, the Peanut Butter Monster Detector may be just the ticket to a good night's sleep for your child (and you!).

WAKE UP! TIME TO CRY!

It's 2:43 A.M. I know this because I can (barely) make out the blurry numbers of the clock radio on my nightstand. I was awakened by tiny ice cold feet and the quiet whimpering of my six year old daughter who has crawled into bed with me. "Daddy" she whispers, "there's a monster in my room!" This is not a completely uncommon happening at my house, and from what I've heard from other parents, not uncommon in their households, as well. For many young children, occasionally being afraid of monsters is a normal part of growing up. Many kids have various "security blankets" that they use to comfort them or dispel their monsters, allowing them to sleep. But some (like my daughter) require a more "active" reassurance. To try and put my daughter's mind at ease (and, consequently allow dear 'ol dad a bit more shut-eye), I decided to take a high-tech approach and see if I could make a device to assist her in learning to manage night time fears by putting her in charge of the solution.



MONSTER THEORY

After giving the problem some thought, I came up with a pretty clear picture of what I wanted to create. The idea was to build something that would double as both an active, immediate anxiety reducer and then act as a standby reassurance system.

It would have to be something easy enough to be operated by a child of five or so, even if she is frightened and in the dark. I figured a large back-lit pushbutton switch should work. When pressed, the unit should instantly brighten the room by switching on high-brightness white LEDs. This would help to immediately lower the anxiety caused by the darkness itself.

Next, the unit should act as a "Monster Detector" by performing some type of active scan accompanied by blinking LEDs and a "scan head" that can show the device is actually doing something. A voice recording could narrate the activities and could be accompanied by some beepy sci-fi sound effects for good measure.

After a scan has completed, the unit should declare the area free of monsters and instruct the child to go back to sleep. It could then slowly dim the white LEDs from very bright, to a very low night-light level, eventually turning off completely. During this fade out period, I decided the device should probably not be re-triggerable. I did this to reduce the possibility that the child would multi-trigger the device using it as a toy or as a distraction from going to sleep. So, now that I had a vision of what I wanted to build, it was time to start on a prototype.

STEP 1: EAT PEANUT BUTTER!

Our family goes through an enormous amount of peanut butter. We seem to always have empty containers and, due to our involvement in Scouting and other school crafting projects, we save all manner of jars, jugs, Popsicle sticks, and the like. While rummaging for a good container for this project, I found the ubiquitous peanut butter jar. Made of sturdy transparent plastic with a large opening and a lid that is easy to remove and replace, it seemed to be a perfect Monster Detector chassis. The 36 oz size jar (Figure 1) has an opening big enough to accommodate some pretty large parts, and the jar itself would make a good acoustic case for the speaker. Some jars have a ridge around the top that would be ideal for holding the white LEDs. The physical size also is about right to sit on a night stand without taking up too much space.

Once the label is removed, the transparent nature of the jar makes it possible to mount all the LEDs inside the jar without having to drill holes. This also leaves all of the circuitry visible while protecting small or fragile moving parts from damage by curious little fingers. If you prefer not to have the circuit boards visible, a paper template could be printed out and then inserted inside of the jar. When back-lit by the LEDs, any printed items would become visible during the scan process and the subsequent monitoring mode of the device. So, now that we have a cheap and useful container, we need the parts to go in it.

TO THE JUNK BOX! AWAY!

I'm a life-long pack rat and I'm usually surrounded by all kinds of interesting bits and pieces (as those of you who read the September '08 Personal Robotics column "Habitat for Hobbies — Part 2" might recall!). So, I began digging through my parts bins looking for the electronic flotsam and jetsam I could cobble together for this device. Experience has shown that a project like this is usually simplest to accomplish by using a microcontroller. The BASIC Stamp 1 OEM version (Figure 4) seemed a good match for this project because:

- It has a built-in LM2940 5V regulator.
- It is easy to program using the simple and mature PBASIC language.
- It has eight I/O pins which is plenty for a small project like this.
- It has a large library of example code and excellent support from Parallax.
- I already had one sitting in my parts bin!

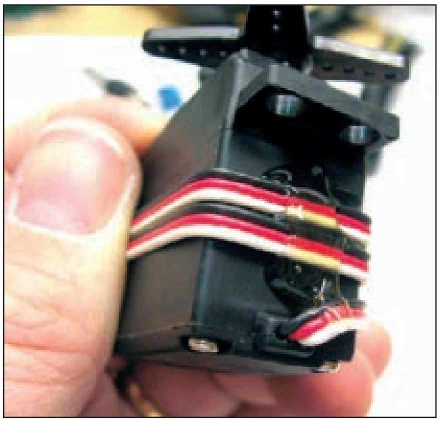
A bit more digging and I was lucky enough to uncover all the things I needed to get started on my prototype (Figure 2) including a nice grid-style perf board, some assorted LEDs, a servo motor, a big lighted pushbutton switch (where I had conveniently replaced the 24V incandescent bulb with a red LED), a low-cost RadioShack sound recorder board, and some white LED Christmas lights. For a comprehensive list of everything, see the Parts List.

WHAT, ME WORRY?

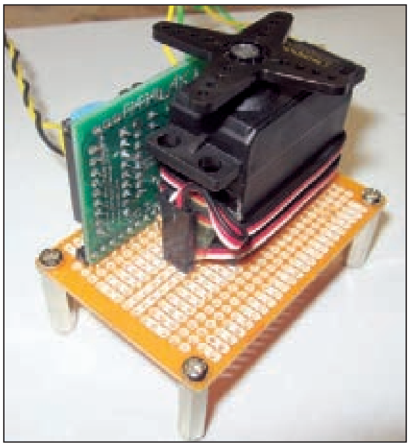
In keeping with my usual style (i.e., "How hard could it be?"), I decided to dive right in and start building. As I was relatively certain of what I was going to do and how the pieces would all fit together, I

■ FIGURE 1. The Peanut Butter Monster Detector chassis.

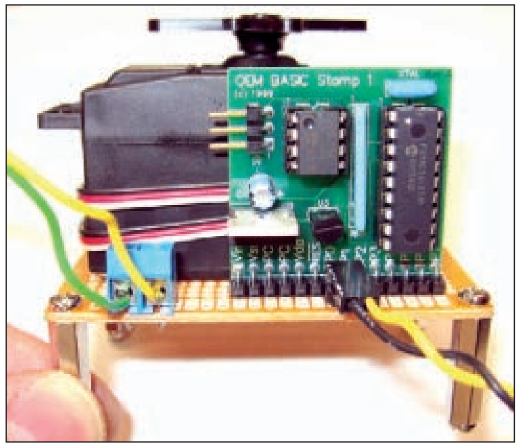
■ FIGURE 2. The junk box components rounded up for the prototype.



■ FIGURE 3. Excess wire wrapped around the servo and hot-glued in place.



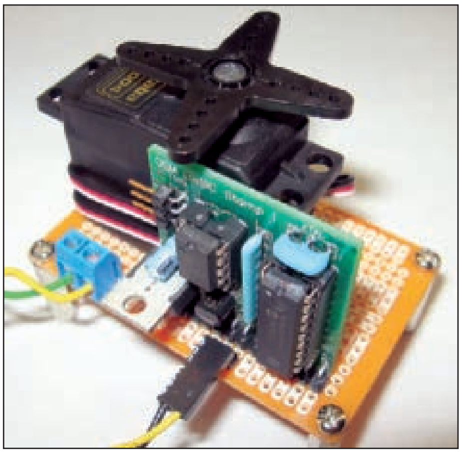
■ FIGURE 4. BASIC Stamp 1 OEM module and servo mounted to PCB.



■ FIGURE 5. Checking the servo motor has clearance to turn above the BS1 OEM board.

skipped breadboarding the circuit and just started soldering the BASIC Stamp 1 onto the perf board. The .100″ pin spacing lined right up and it was nice and solid once soldered. I then took the servo motor and wrapped the excess lead length around the body (Figure 3) creating a

■ FIGURE 6. Screw terminals for power and headers for pushbutton/LED connections.



nice "candy stripe" look. I then hot-melt glued the servo to the perf board (Figure 4) after checking to be sure that with a servo horn attached, it would clear the top of the BASIC Stamp board (Figure 5). I flipped the board over and added a three-pin .100″ header for the servo, a 90 degree, right-angle two-pin header for the pushbutton and the pushbutton's LED, and a pair of screw terminals for power input (Figure 6). To make the board easier to mount in the jar, I added some metal standoffs to its four corners.

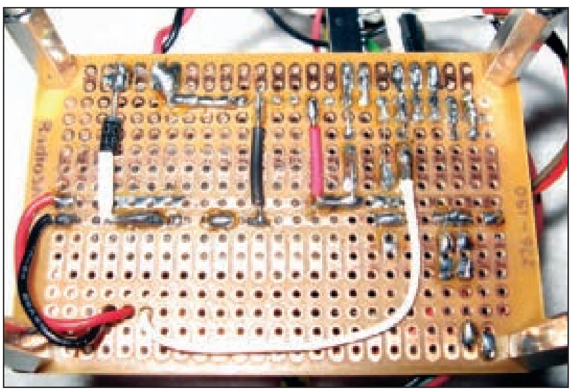
The grid-style PCB (printed circuit board) worked well for this design as I could use the strips that go down the center as a power bus; the pads at the edges let me bring out the pins from the Stamp 1 to the edge (Figure 7). As the BASIC Stamp OEM board has a full LM2940 regulator on it, I decided to use it to regulate power for the entire project. This way, the Detector could be powered by a six volt battery pack or by a higher voltage external power supply.

If you decide to build the unit using a standard BASIC Stamp 1, I suggest you add an LM2940 voltage regulator to provide the V+ as this way you won't have to worry about destroying components if the wrong voltage level is accidentally applied. Also, if you look closely at the solder side (Figure 7) you will notice I included a diode in series with the power input as insurance against improper polarity. I took these extra steps since I knew the unit would be used around children and there was a good chance the wrong wall wart might be used to power the unit by accident.

MOUNT 'EM UP!

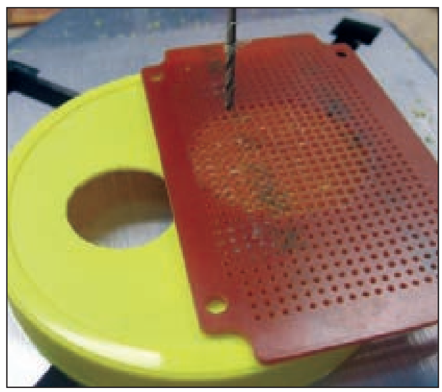
Now that I had the PCB mostly settled, I wanted to get started on the top of the enclosure. It was fairly easy to use a Dremel to cut a hole for the pushbutton switch (Figure 8), but the next step would require some pretty

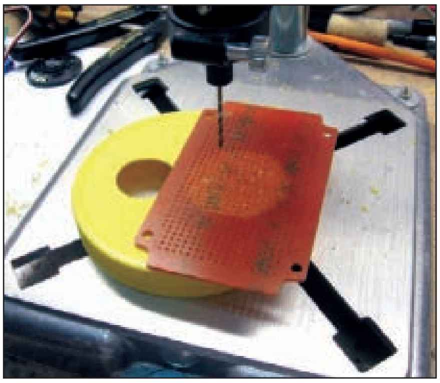
■ FIGURE 7. Solder-side view of the PCB with all wiring complete.



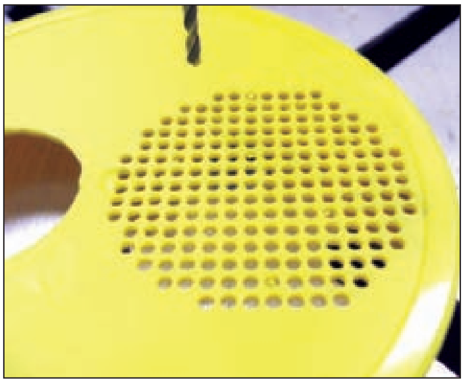
■ FIGURE 8. Lighted push-button mounted in jar lid.

■ FIGURE 9. Old PCB hot-melt glued to lid top as a drill template.





■ FIGURE 10. Dremel drill press used to drill holes for speaker grill.



■ FIGURE 11. Speaker grill after PCB drill template is removed.

■ FIGURE 12. Speaker hot-glued to the bottom side of the jar lid.

precise drilling in order to get a nice professional look. Since there was going to be a speaker in the jar lid, there would have to be holes to let the sound out. Though I could have just randomly drilled a lot of holes, I didn't want the unit to look quite that rough. So, to get nice, even spacing, I took an old piece of .100" perf board and hot-melt glued it to the top of the jar (Figure 9). I then placed the lid on my Dremel drill press and used the perf board to guide the bit as I punched holes to make a nice circular grid (Figure 10). When I was done, I popped the old PCB off the top leaving a nice speaker grill (Figure 11).

Next, I hot-melt glued the speaker that came with the 20 second sound module to the underside of the jar lid (Figure 12). I placed the pushbutton back into its hole (Figure 13) and then did a test-fit on the top of the jar (Figure 14). I put the main PCB with the servo on it down inside the jar and marked four dots on the bottom that corresponded to the locations of the four standoffs. I drilled four holes and ran in a couple of screws to hold the PCB in place to see how I was for clearance from the servo horn to the pushbutton body (Figure 15). It fit with room to spare! Now that I had a button and a speaker, it was time to get the sound module working.

■ FIGURE 13. Pushbutton test-fit next to speaker.

SOUNDING OFF!

The RadioShack 20 second sound module is a cute little device that really lends itself to hacking. Though the sound quality isn't terrific, it's usually good enough to playback voice quality recordings. Strangely, the external pushbutton switch is the one used for recording, the playback button is located on-board (seems sorta backwards to me!). Through some trial and error, I discovered that the onboard playback pushbutton can be remotely operated by adding a wire to the top resistor pad of R3 as shown circled in red in Figure 16. When this wire is taken to ground and then released, the unit will begin to play back its stored sound.

I hot-melt glued the sound module to the opposite side of the servo motor on the PCB and then ran the power

MONSTERS DETECTED! RUN FOR YOUR LIFE!

Inevitably, when I've described the operation of the detector to friends and colleagues, someone always asks "can you set it so that once out of every three or four activations, the unit sounds a siren and yells 'MONSTERS IN THE CLOSET! RUN FOR YOUR LIFE!!"

Though we all got a chuckle out of imagining the result of such a scan, in reality I have never considered actually doing this, even as a joke. Some folks also asked "Could you have the unit periodically declare a monster is present on the first pass and then, after the second pass, declare the monster has left?"

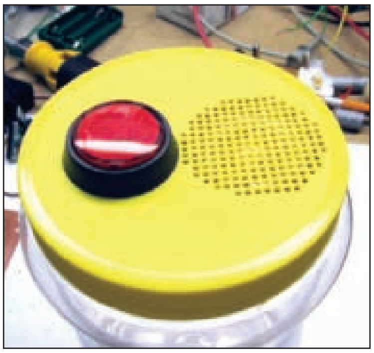
I considered the dialog very carefully and even went so far as to consult some friends in child care and child development to insure that the dialog had the intended purpose. In the end, I decided that the Peanut Butter Monster Detector should never detect a monster. If it ever gave a "positive" reading, that would tend to lend credence to the concept that imaginary monsters exist OR that the detector could be "fooled". If you think about it, giving a false positive leads to the question, "Could it give a false negative?"

For maximum reassurance, not only does the unit NOT detect a monster on the first pass, it then double-checks to make sure that the first pass was not mistaken. Not only is this doubly reassuring, this also causes the unit to make sounds and perform actions for about 20 seconds or so, helping the child focus on the device and forget about whatever noise or feeling awoke them in the first place.

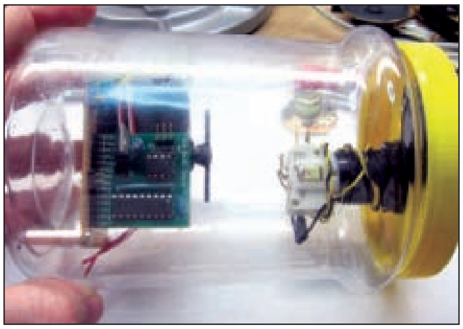
So far, this approach has worked very well to provide reassurance and to empower my child by providing a way for her to take charge of her own fears and learn to help herself sleep through the night.

So, if you were thinking of writing to ask me about this, no you're not the only person who's thought of it.

Also along these lines, when you are recording the dialog for your detector, make sure you are alone and that any older siblings aren't around to see how you do it. Recording over your reassuring words with "GIANT MONSTER DETECTED UNDER THE BED! RUN FOR YOUR LIFE!" is just the sort of trick older siblings might love to play on the younger ones! You have been warned!



■ FIGURE 14. Lid test-fit back on the jar.



■ FIGURE 15. Test-fit for clearance from the servo horn to the bottom of the button.

■ FIGURE 16. Add a trigger wire to resistor pad R3 (circled in red).

leads down through the PCB to the center power bus. I ran the speaker leads to a two-pin header that would lead up to the speaker in the jar lid. I threaded the "trigger" wire from the R3 pad down through the PCB and across to the P3 pad for the BASIC Stamp 1 (Figure 17). I hot-melt glued the record pushbutton to the back of the servo motor (Figure 18). As a quick test, I recorded a "hello world" message and then wrote a small program for the BS1 to toggle the pin. I found a dwell of about 500 ms was perfect to get reliable triggering of the sound sample. So far, so good!

JUNK BOX SCAN HEAD

So now that I had sound, I started to think about the scan head that the servo motor would be rotating. I dug around in the junk box and came up with an old prescription pill bottle with a child proof cap. I cut around the bottle about 1/2" from the top with a Dremel tool and then drilled holes at four cardinal points to hold the LEDs (Figure 19). I used two small screws to secure the pill bottle to the servo horn (Figure 20), hot-melt glued a small tire hub from a discarded toy to the top (Figure 21), and voíla! I had my high-tech scanner head.

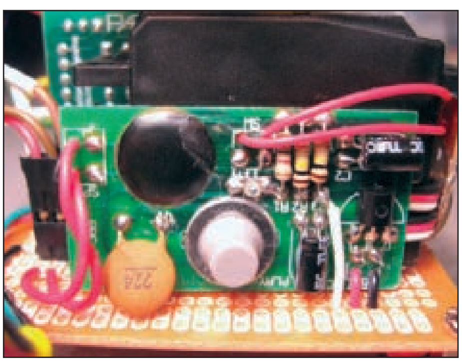
MIXING IT UP

When I was sketching up the original plans for this device, I had intended to have both the audio playback from the sound recorder and the BS1 beepy sound effects both coming from a single speaker. I figured I could just force feed the sound from the BS1 to the sound board's speaker by loosely coupling the output pin from the BS1 to one of the speaker leads using a small capacitor.

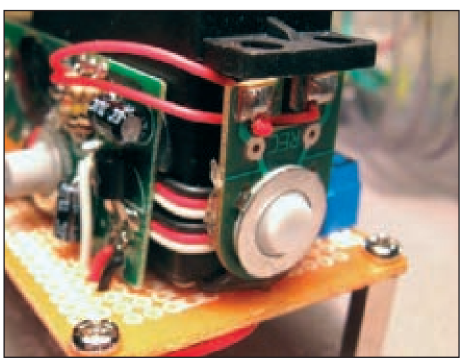
When I tried this, the volume from the sound module dropped drastically and became very distorted. I tried a few other methods to mix the signal from the sound playback board and the output from the BS1 including a small transformer and even a couple op-amps to make a small mixer. The circuit was getting more complicated and the results were still pretty dismal. I decided to take a simpler approach and create a mechanical mixer.

Instead of trying to mix two signals into one speaker, I placed a second speaker coaxially mounted on top of the first speaker! I did this by cutting a ring from the remains of the pill bottle I had used earlier for the scan head, and then hot-melt gluing it to the existing speaker. I then glued a small surplus speaker I found in my junk box to that ring (Figure 22). This method worked very well and made the BS1 sound effects audible right along with the spoken dialog from the sound module.

This also meant that it would be possible to build this device with a second speaker grill, maybe even facing in a different direction. This way, the beeping effects could appear to come from the scan head while the dialog from the sound module could come from near the main LED light source. You could even consider mounting the second speaker on the scan head itself so the sounds would move around as the direction of the scan head changed. Of course, the path I took isn't a requirement; you can construct your monster detector in any container and with any method you like. Make

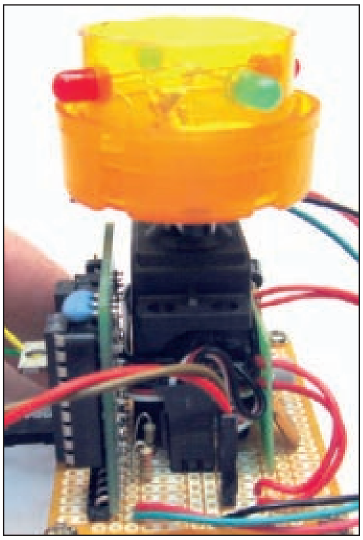


■ FIGURE 17. Trigger wire from pad R3 through the PCB to BS1 pin P3.

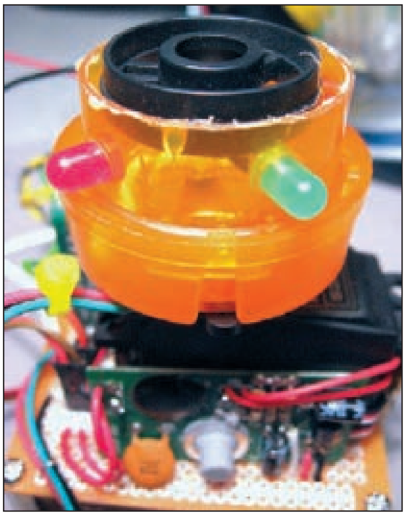


■ FIGURE 18. The record button hot-glued to side of servo.

■ FIGURE 19. Scan head with LEDs mounted and servo horn attached.



■ FIGURE 20. Scan head attached to the servo motor.



■ FIGURE 21. Tire hub from an old toy glued to the top completes the scan head.

■ FIGURE 22. Second speaker piggy-backed on top of first speaker creating a "mechanical mixer."

the "Cigar Box Monster Detector" or the "Cookie Tin Monster Detector" for example. Just follow the schematic (Figure 23) and place the components however you prefer.

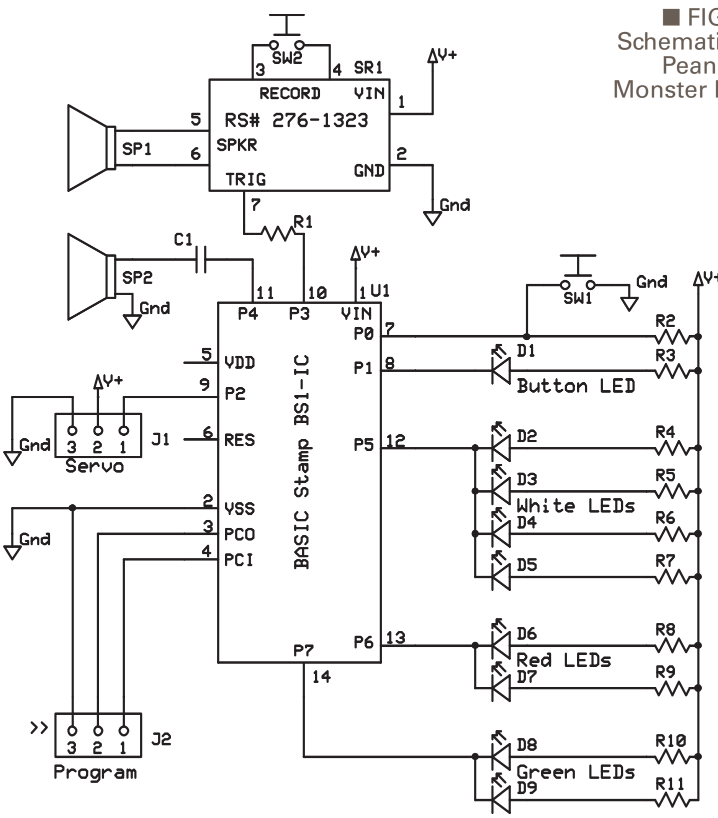
CODING IT UP

Now that I had the hardware done, it was time to pull up the BASIC Stamp programming environment and start to construct the code. Though its considered a bit "long in the tooth" by some and many other micro-controllers surpass it in speed and command set flexibility, there is still a surprising amount of power in the venerable BASIC Stamp 1. For example, I was able to take advantage of PWM output for operating the servo motor and to fade the LEDs, use constants to hold the servo START and END positions, and the random number generator to create those vintage science fiction sounds. Though the complete source code is available on the website (**www.nutsvolts.com**), I wanted to take a moment to go over the basic functions here so you can better understand how the code works and to help you modify it to make it operate the way you want.

A PIN BY ANY OTHER NAME ...

I started the coding by defining all the pins at the top of the program. This way, I could use more human readable code statements. I find it much harder to understand code such as "IF X = Y THEN GOTO Z" as opposed to "IF BUTTON = PUSHED THEN GOTO PLAYSOUND." In keeping with this idea, I allocated the pins of the BASIC Stamp 1 with what I felt were useful names:

- PIN0 – Trigger (N.O. pushbutton to GND pulled "up"via R2)
- PIN1 – LEDTrigger (LED inside pushbutton)
- PIN2 – Servo (servo motor)
- PIN3 – SoundBoard (sound module trigger)
- PIN4 – Speaker (audio out to speaker)
- PIN5 – LEDWhite (white LED string)
- PIN6 – LEDRed (red LEDs in



■ FIGURE 23. Schematic for the Peanut Butter Monster Detector.

SCHEMATIC PARTS LEGEND

■ U1	BASIC Stamp 1
■ SR1	Sound recording module
■ SW1	N.O. momentary lighted button
■ SW2	N.O. momentary button (included w/SR1)
■ SP1	8 ohm speaker (included w/SR1)
■ C1	2.2 µF decoupling capacitor
■ D1	Red LED (internal to SW1)
■ D2-5	White LEDs
■ D6-7	Green LEDs
■ D8-9	Red LEDs
■ R1	100K resistor
■ R2-10	330 ohm resistors
■ J1	.100” three pin header
■ J2	.100” three pin header

scan head)

- PIN7 – LEDGreen (green LEDs in scan head)

The program starts by initializing a few things such as turning on the LED in the pushbutton and then playing a beep through the speaker to let you know it has woken up and is ready to work. Next, the system goes into a loop where a few things happen. First, PWM is used to dim the LED in the pushbutton to a low level so the button doesn’t light up the room. This setting is adjustable in the software so you can modify it to taste. Second, the loop generates a random number for each pass so that when the button is pressed, a new “seed” is used making sure that the random numbers used for the sound effects are more truly random and less pseudo random. Last, the loop checks to see if the pushbutton has been pressed. These three tasks continue indefinitely until the pushbutton is detected.

When the button is pressed, we drop out of the loop and do two things immediately. First, the LED in the pushbutton is turned on to full brightness; second, the white LEDs are turned on. Once this is accomplished, a short acknowledge sound is played out of the sound effect speaker by running through a FOR loop that plays 32 notes for only 1 ms each. This makes an upward sounding “Zrrrrp!” audio effect. After this sound effect is played, the red and green LEDs on the scan head are both switched on.

At this point, the Stamp toggles the sound module trigger line by taking it LOW for 500 ms, then bringing it back to HIGH. This starts the sound module playing back the narration (see sidebar for the dialog I used in my Monster Detector). At this point, both the red and green LEDs on the scan head are turned off and we drop to the first Scan routine.

Scan #1 is a FOR loop that steps from the START position of the servo (0 degrees) to the FINISH position for the servo (180 degrees). The servo should have a pulse sent to it every 20 ms in order for it to continue to move but, rather than placing a 20 ms pause command in the loop (as is typically done), I drop to a subroutine that plays random tones through the speaker. After the subroutine returns, I toggle the state of the red LED in the scan head. This makes the red LED strobe on and off as the servo pans the head to enhance the scanning

■ FIGURE 24. Sami Graner fast asleep next to her Peanut Butter Monster Detector.

■ FIGURE 25. The Peanut Butter Monster Detector featured on KOOP radio in Austin, TX.

PARTS LIST

- ■ 1 BASIC Stamp 1 — Parallax BS1-IC
- ■ 1 BASIC Stamp 1 Serial Adapter — Parallax (item code 27111)
- ■ 1 Hobby servo motor — Parallax (900-00005) or Futaba (3004)
- ■ 1 20 second sound module — RadioShack (#276-1323)
- ■ 2 Green LEDs and 2 red LEDs — RadioShack Assorted LEDs (#276-1622)
- ■ 1 Grid-style PC board — RadioShack (#276-150)
- ■ 12 330 ohm resistors — RadioShack Five-pack (#271-1315)
- ■ 1 100K resistor — RadioShack Five-pack (#271-1347)
- ■ 1 Metal standoff with screws (four-pack) — RadioShack (#276-195)
- ■ 1 N.O. pushbutton switch — Happ Controls (D54-0004-20) or All Electronics (LPS-1R)
- ■ 4 White LEDs — All Electronics (LED-75)
- ■ 1 .100” snappable header — All Electronics (SHS-32)
- ■ 1 Empty peanut butter jar (yum!)

RESOURCES

■ Parallax
www.parallax.com

■ All Electronics
www.allelectronics.com

■ Video of the Peanut Butter Monster Detector
www.youtube.com/VernGraner

■ Interview with Samantha Graner:
www.notepad.org/SamiPBMD.mp3

effect. These commands take approximately 20 ms to complete, so the pulses arriving at the servo are timed about right. When Scan #1 has completed, I call a subroutine (called “BLINKY”) that uses a FOR loop to light the red and green LEDs alternately for 200 ms each, making it appear the unit is evaluating the results of the scan. When the BLINKY routine is complete, the green LED is lit and the BASIC Stamp plays a high-pitched tone to indicate the results of the scan have been evaluated. It then lights the green LED to indicate that all is well. At this point, another scan is initiated but with the START and FINISH values of the servo reversed so that the scan head returns to its starting position. The same lights and sounds are played and at the end of the second scan, the green LED is again lit and then it blinks on and off at half second intervals for five seconds to draw attention to the fact that the scan result has not detected any monsters. With the second scan completed, the red LED button is turned off and then the unit goes into another FOR loop. A PWM command is used in the FOR loop to slowly fade the white LEDs from full brightness to completely out. During this time, the unit will not respond to another button press (this could be changed if desired). Once the white LEDs are completely off, the unit jumps back to the top where it places the pushbutton LED in a dim glow and is again ready to start a scan when a button press is detected.

PEACEFUL, EASY FEELING

When my daughter goes to bed, she insists on having her Peanut Butter Monster Detector on her bedside table. Recently, we moved to a new house and her bedroom wasn’t quite ready yet. She opted to sleep on the sofa, but only if I put the Detector on the end table so she could reach it, just in case (Figure 24).

MONSTER DETECTOR SCRIPT

"Monster Detector Active. Scanning ..."

(pause 5 seconds)

"No monsters detected. Scanning again ..."

(pause 5 seconds)

"All clear! No monsters detected in vicinity!"

"Go back to sleep"

"All Clear"

"Go back to sleep"

The unit has been quite a hit with her and she enjoys telling people how it works (check the resources section for an audio interview where she describes the detector herself). The Peanut Butter Monster Detector was even featured on a local radio program where it got its own chair and microphone (Figure 25)!

THE MONSTROUS CONCLUSION

If you're interested in making your own Monster Detector, but maybe don't have a junk box as well stocked as mine, fear not! I've been kicking around the idea of putting together a kit. No definite plans as of yet, but it will be very similar to the project in my article. You might want to stop by http://store.nutsvolts.com once in a while and see what's new! In the meantime, I'd love to hear from anyone who decides to build their own Monster Detector. As always, if you have questions or comments drop me an email at vern@txis.com. Have a safe and Happy Halloween! **NV**

THANK YOU

I want to thank my daughter, Samantha, for helping inspire and test the Peanut Butter Monster Detector, and my son, Nicholaus, for lending me the small plastic monsters we used for testing the Detector. (*Of course* it didn’t detect them because they weren’t *real* monsters! Duh!). A special thanks to John Richter for drawing the Peanut Butter Monster Detector logo/mascot. Also, a special thank you to Robin Lemieux for her patience and persistence ... without which this article would never have seen print!

■ BY FRED EADY

ROLL YOUR OWN WIFI HOTSPOT

THIS MONTH, WE WILL BUILD a very useful data communications device that is right at home with just about any of today's laptops, PDAs, and personal computers. Depending on your intended application, our easy-to-build PIC-based data communications device can transfer data using its serial port or its WLAN (Wireless Local Area Network) capabilities. If you prefer to exclude Windows and Linux from your data communications network, the device we're about to discuss also has the ability to communicate and transfer data wirelessly with another like device.

The technology behind our little device is not at all new. Everything WLAN about our PIC peripheral is based on the 11 Mbps IEEE 802.11b wireless standards. In fact, the data communications device we will be assembling has roots in an earlier PIC-based WLAN device that was called the AirDrop-P. There's even an 802.11b book that you can reference that explains every detail of the WLAN firmware driver and hardware we're about to construct. The best thing about this device is its cost, which is less than one half of the cost of the old AirDrop-P.

FROM THE ASHES

...rises the WLAN Phoenix. The WLAN Phoenix is a smaller, more compact version of its predecessor, the EDTP AirDrop-P. Just because the WLAN Phoenix is smaller, doesn't mean it is less powerful. The Phoenix uses the same 11 Mbps CompactFlash 802.11b radio technology that was part of the AirDrop-P. The Phoenix firmware driver is designed to enable any CompactFlash 802.11b radio card that is based on the PRISM 1, 2, or 3 chipset. The Phoenix firmware driver will not activate PRISM GT-based CompactFlash radio cards. The firmware driver will also enable PRISM-based PCMCIA cards, as well as CompactFlash technology that was spawned from PCMCIA technology.

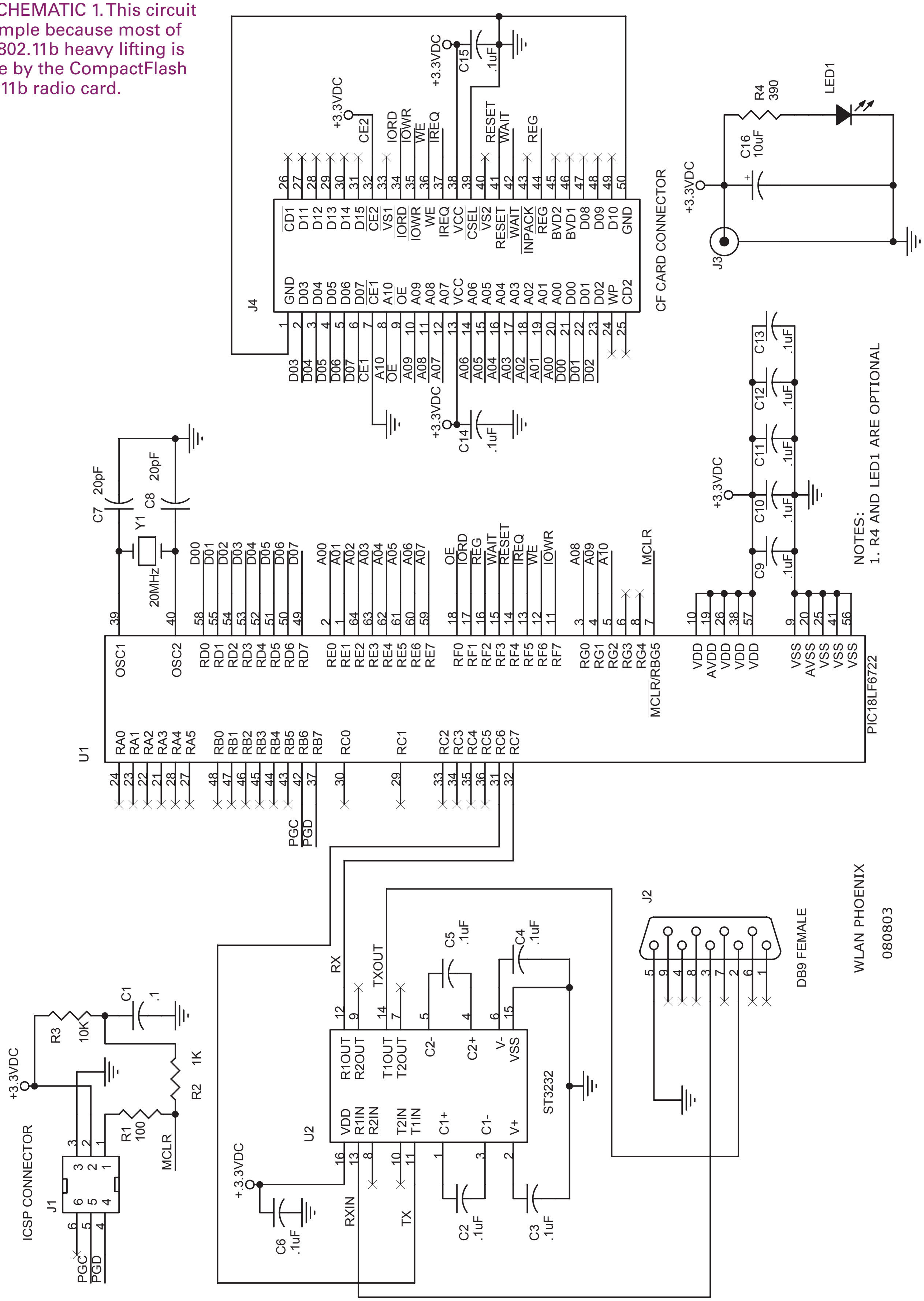
To make the Phoenix easier to put into service, I will provide a compatible PRISM-based CompactFlash radio card for those of you that cannot obtain one through your normal distributor channels. The Phoenix can operate with the NetGear MA701 CompactFlash radio card, the D-Link 660W radio card, the Zonet ZCF1100 radio card, and the LinkSys WCF12 radio card. An Xterasys PRISM-based CompactFlash 802.11b card with the FCC ID of MQ4CWB1K also works with the Phoenix.

If you search diligently, you can find the 802.11b CompactFlash 802.11b radio cards I've mentioned from various vendors on the Internet. A list of PRISM-based 802.11b cards can be had from the Linux open source 802.11b pages also on the Internet. The PRISM chipset-enabled Xterasys 802.11b CompactFlash 802.11b radio card I am providing for you is shown in Photo 1.

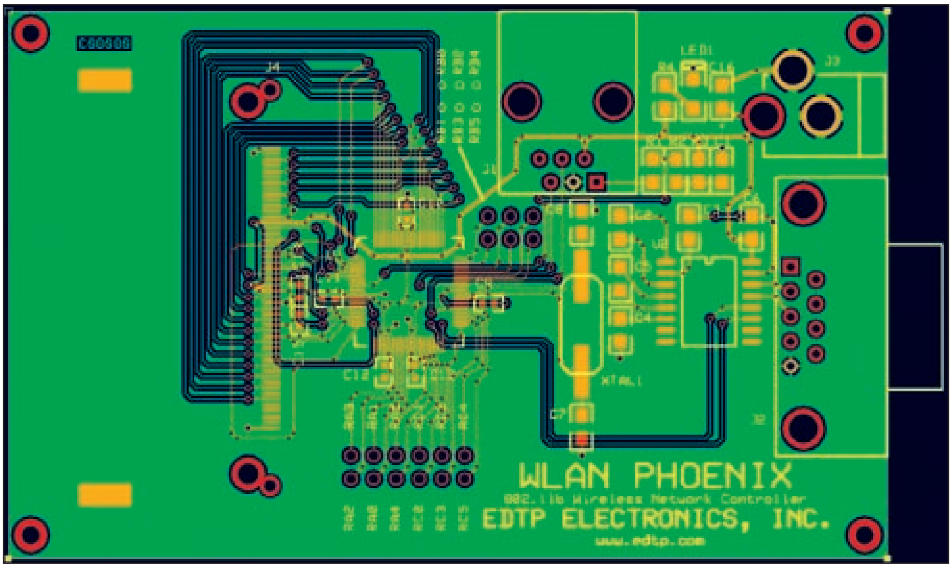
WLAN PHOENIX HARDWARE

With the exception of the free PIC18LF6722 I/O lines, the PIC18LF6722 is dedicated to serving the CompactFlash socket and the serial port. If you're wondering if you can do other things with this CompactFlash circuit outlined in Schematic 1, the answer is yes. In fact, an AirDrop user donated some source code to read and write CompactFlash memory cards using this hardware configuration. If you're interested in CompactFlash memory card access, you can get the source code from the download area of the EDTP Electronics, Inc., website (**www.edtp.com**). Within this text, we're going to focus on getting the Phoenix online with the 802.11b CompactFlash radio card.

■ PHOTO 1. This is the Xterasys CompactFlash card I mentioned in the text. This card's lineage reaches back to PRISM-based 802.11b CompactFlash cards offered by Trendnet.



■ SCHEMATIC 1. This circuit is simple because most of the 802.11b heavy lifting is done by the CompactFlash 802.11b radio card.



■ SCREENSHOT 1. The WLAN Phoenix is easy to assemble due to its small number of parts. Patience and standard electronic component soldering tools are all you need to put your own WLAN Phoenix online.

The RS-232 port was originally designed into the WLAN Phoenix as a debugging device. If you browse through the Phoenix source code, you'll find lots of status and debugging messages that are commented out. Most of the deactivated messages are aimed at the RS-232 port, which was most always connected to a Tera Term Pro terminal emulation window when I was developing the Phoenix firmware.

One of the primary reasons for bringing the Phoenix to life lies in the fact that I continually get email from readers requesting information on how to build a simple device that can act as a RS-232 to wireless Ethernet bridge. In addition to its EUSART, the PIC18LF6722 contains two additional internal serial communications engines, which allow the Phoenix to also act as an 802.11b communications bridge for I^2C and SPI protocols. I^2C and SPI drivers are not present in the current version of the WLAN Phoenix firmware driver. However, the hardware I/O pins to implement I^2C and SPI protocols are free and available to the Phoenix user that wishes to add either of these capabilities.

The architecture of the WLAN Phoenix driver source code — particularly the wlan_phoenix.h include file — is a direct result of the CompactFlash connector pinout. Think of the CompactFlash connector as a portal to an intelligent and interactive microcontroller peripheral. A standard parallel memory device such as an SRAM (Static Random Access Memory) simply places what you deposit on its data bus pins in the location you have specified on its address pins. With the activation of the read control line, that same SRAM retrieves the data from the memory location pointed to by the address lines. Other than presenting data to the requestor on its data bus, the SRAM doesn't talk back in an interactive manner. It just does what it was designed to do and no more.

You'll find that the CompactFlash 802.11b radio communicates with you depending on what you place on the CompactFlash connector's data bus, address, and control pins. To utilize the capabilities that are built into the CompactFlash card, we must ask it questions and use the answers to get at the particular internal features of the 802.11b card that we want to employ. I've left the powering of the Phoenix up to the user. Some of you may use battery power and some of you may plant the Phoenix in a permanent location that has access to a suitable power source. Be sure that your WLAN Phoenix power supply can supply at least 300 mA, which is the maximum current drawn by the Xterasys CompactFlash 802.11b radio card while transmitting. The Xterasys radio card I am supplying should only be powered with 3.3 VDC. It's specifications state that you can also power the CompactFlash radio that I'm using with 5.0 VDC. I've never tried to do that, so, you're on your own if you want to go there with the Xterasys radio card.

The Phoenix is designed to be very easily built from scratch. Its printed circuit board (PCB) is a standard two-layer ExpressPCB production board that houses a mere 24 electronic components. At first glance, the CompactFlash connector looks to present the most assembly difficulty. However, you'll find that with care and the right tools, the connector is one of the easiest parts to place and solder down to the PCB. I used a base-model Metcal soldering system and solder paste to plant the connector you see in Photo 2.

I also tested a Xytronic hand-held hot air wand on the CompactFlash connector's pins. The hot air wand "blew up" some really pretty solder joints on the connector's fragile pins. The hot air soldering station I used can be purchased from Jameco as part number 262705. I highly recommend the Xytronic hot air soldering station as it has saved my butt time and time again. It is indispensible for both SMT repair and SMT prototyping work.

WLAN PHOENIX FIRMWARE DRIVER

It is not necessary to completely understand every line of code that forms the fabric of the WLAN Phoenix firmware driver. All of the details of initializing the CompactFlash 802.11b radio card, sending data, and receiving data have been addressed and organized into easy to understand user-called functions. If you have a thirst for the minute details of how the firmware driver operates, I suggest getting yourself a copy of the book *Implementing 802.11 with Microcontrollers*. The book goes into nauseating bit-by-bit detail as to how the Phoenix firmware works. You can also get lots of good information by studying the Linux 802.11b open source driver code, which is available from various points on the Internet. The WLAN Phoenix firmware was modeled from the open source Linux 802.11b source code.

The CompactFlash radio card is very

smart once properly enabled. Before any of its WLAN circuitry can be used, it must be initialized. The WLAN Phoenix firmware performs the radio card initialization with a call to the init_cf_card function. The init_cf_card function also performs I/O port setup for the PIC18LF6722.

Recall that the CompactFlash 802.11b radio card must be asked certain questions to obtain answers we need to use its feature set. CompactFlash and PCMCIA cards contain a memory area that is made up of individual memory elements called tuples. The init_cf_card function enters the tuple memory area and examines each tuple and its link. All of the tuples are arranged as a linked list of memory elements. Link tuples point to the next available data tuple. The init_cf_card function flows through the path laid out by the link tuples, picking up data from each relevant data tuple along the way.

There are many types of tuples that provide unique information we will need to successfully use the CompactFlash 802.11b radio card. The idea behind navigating the card's tuple memory is to glean the address we need to issue commands to the radio card's PRISM controller. Once the init_cf_card function has navigated to and retrieved the COR (Configuration Option Register) address tuples, it issues an initialization command to the CompactFlash 802.11b radio card by way of the COR.

Once the CompactFlash 802.11b radio card is successfully initialized, we can issue configuration commands to the radio card's PRISM engine. These configuration commands are passed via the phoenix_cfg function. Configuration options such as what type of network the card will operate on, the speed in which it will operate, and whether or not to employ WEP (Wired Equivalent Privacy) encryption are set using the phoenix_cfg function.

The Phoenix is capable of operating in a standard WLAN BSS (Basic Service Set) environment that includes a wireless router or AP (Access Point) in the network. There's a very good chance that your home WLAN is configured as a BSS. The Phoenix can also operate in a peer-to-peer IBSS (Independent Basic Service Set) network that does not include a router or access point.

By feeding the phoenix_cfg function the right diet of bytes, we can force the CompactFlash 802.11b radio card to run at 1 Mbps, 2 Mbps, 5.5 Mbps, or 11 Mbps. We can also specify the SSID (Service Set Identity) of the network we wish to join with the desired SSID argument string and a call to the phoenix_cfg function. For humans, the SSID is the ASCII name for the network. The SSID can consist of up to 32 bytes. The WLAN Phoenix firmware driver comes coded with PHOENIX_ NETWORK as its desired SSID.

In addition to user-friendly initialization and configuration functions, the WLAN Phoenix firmware driver also contains built-in IP (Internet Protocol) functionality that is also just a function call away. The Phoenix can converse using IP-based protocols such as UDP (User Datagram Protocol) and TCP (Transmission Control Protocol). An active WLAN Phoenix also has the ability to respond to a PING, as an ICMP (Internet Control Message Protocol) echo routine is built into the driver. To facilitate establishing communications with other network nodes, the WLAN Phoenix firmware driver contains a built-in ARP (Address Resolution Protocol) engine that is able to respond to and issue ARP messages.

UDP communications are the easiest of the IP protocols to understand and implement. So, the firmware driver includes an RS-232 based demo utility that allows the user to define local and remote IP addresses and UDP port numbers. The WLAN Phoenix user can then use the UDP demo utility to send eight bytes of user-defined UDP data to the remote node.

Fortunately, not every nuance of the wireless Ethernet network must be controlled by the WLAN Phoenix firmware driver. For instance, all of the network link management is performed automatically by the CompactFlash 802.11b radio card.

ASSEMBLING A WLAN PHOENIX

The ExpressPCB printed circuit board layout of the WLAN Phoenix is depicted graphically in Screenshot 1. If you exercise patience, you can easily mount and solder all of the electronic components and connectors onto the pads you see in Screenshot 1 with standard soldering tools. If you decide to use spool solder, get the smallest diameter of solder that you can. That will help eliminate the creation of large solder blobs on the CompactFlash connector pins. I suggest using solder paste (like the type that is available from Stencils Unlimited) and a hot air soldering system for the CompactFlash connector and the PIC18LF6722. The hot air will give you professional-looking solder joints and you will avoid the solder blob problem. Be careful not to overheat the components with the hot air. Don't get in a hurry and use just enough heat to flow the solder paste.

Although I'm offering the WLAN Phoenix as a complete kit of parts, some of you may want to build your own WLAN Phoenix entirely from scratch

■ PHOTO 2. Twenty four easy-to-mount electronic components and four connectors is all it takes to put the WLAN Phoenix on the air.

```
WLAN PHOENIX UDP UTILITY Ver 8.09.08

    1   CYCLE TIME (SECONDS) - 1
    2   PHOENIX IP ADDRESS   - 192.168.1.150
    3   REMOTE IP ADDRESS    - 192.168.1.100
    4   PHOENIX UDP PORT     - 8088
    5   REMOTE UDP PORT      - 30303

    PRESS ESC TO RUN

    REMOTE Hardware Address = 00-00-00-00-00-00
    LOCAL Hardware Address  = 00-00-00-00-00-00
```

■ SCREENSHOT 2. The WLAN Phoenix firmware driver is hard-coded to receive UDP packets via port 8088 decimal. As you can see, I've entered the IP addresses and the UDP port numbers to allow the WLAN Phoenix to communicate with a PC using UDP.

```
WLAN PHOENIX UDP UTILITY Ver 8.09.08

    1   CYCLE TIME (SECONDS) - 1
    2   PHOENIX IP ADDRESS   - 192.168.1.150
    3   REMOTE IP ADDRESS    - 192.168.1.100
    4   PHOENIX UDP PORT     - 8088
    5   REMOTE UDP PORT      - 30303

    PRESS ESC TO RUN

    REMOTE Hardware Address = 00-00-00-00-00-00
    LOCAL Hardware Address  = 00-00-00-00-00-00
```

■ SCREENSHOT 3. When things go right, the WLAN Phoenix firmware driver collects the CompactFlash 802.11b radio card's hardware (MAC) address, as well as the remote node's hardware address (which was gleaned from the WLAN Phoenix's ARP request).

using a PCB and parts you purchase from the distributors that advertise in *Nuts & Volts*. To aid you in getting a scratch-built WLAN Phoenix online, I'll provide the ExpressPCB file with the WLAN Phoenix download package, which you can get from the *Nuts & Volts* website.

As the assembly of the WLAN Phoenix is rather obvious, let's specify the components you'll need to obtain to scratch build a WLAN Phoenix. The 50-pin CompactFlash connector is a Molex product and is available from Digi-Key (part number WM18587-ND). The 0.1 µF power supply bypass capacitors (C9-C15) that support the PIC18LF6722 (Digi-Key PIC18LF6722-I/PT-ND) and the CompactFlash connector are packaged in 0603 format. I like to use X7R ceramic capacitors for general-purpose work. So, the 0.1 µF 0603 capacitors I chose to use in the original WLAN Phoenix hardware are Digi-Key part number PCC1762CT-ND.

The 0.1 µF bypass and charge pump capacitors (C2-C6) that support the STMicroelectronics ST3232C (Mouser 511-ST3232CWR) RS-232 interface IC are packaged in 0805 format. The same 0805 0.1 µF capacitor used to support the ST3232C is also used in the PIC18LF6722's MCLR circuit (C1). The 0805 form factor 0.1 µF capacitors I use in the WLAN Phoenix hardware are Digi-Key part number PCC1812CT-ND. The 20 pF 0805-packaged capacitors that support the 20 MHz crystal (Digi-Key XC581CT-ND) are 5% tolerance C0G types (Mouser 80-C0805C200J5G). It's always a good idea to post a capacitor at the PCB power input when the power is not generated and regulated on the board itself. So, I put a 0805-packaged 10 µF tantalum (Digi-Key 511-1447-1-ND) on guard duty behind the power input receptacle, J3 (Digi-Key CP-202A). The connection to your PIC debugger/programmer is provided by a Tyco Electronics/AMP six-pin modular jack, which can be purchased from Mouser as part #571-5520250-3. Jameco provides the RS-232 female connector as part number 104951.

I always use 1% resistors in my projects. However, you can substitute 5% values in the Phoenix circuitry as the resistors are not used in critical circuits that require tight resistance tolerances. I also tend to load 1206 form factor LEDs onto my electronic creations. Any 1206 LED that suits you may be used in this application.

```
C:\Microchip Solutions\Microchip\TCPIP Stack\Utilities\MCHPDetect.exe
Microchip Announcement Detector
Copyright (c) 2004 Microchip Technology, Inc. Ver. 1.0 (Oct 06 2004)
Written by Howard Henry Schlunder

Waiting for data on UDP port 30303...

Total packets received: 1
Received 8 bytes of data from 192.168.1.150
12345678

Total packets received: 2
Received 8 bytes of data from 192.168.1.150
12345678

Total packets received: 3
Received 8 bytes of data from 192.168.1.150
12345678

Total packets received: 4
Received 8 bytes of data from 192.168.1.150
12345678

Total packets received: 5
```

■ SCREENSHOT 4. This should not be a surprise in content or existence.

RUNNING THE UDP DEMO UTILITY

The WLAN Phoenix firmware driver that you download from the *Nuts & Volts* website (**www.nutsvolts.com**) contains an application that is hard-coded to transmit eight bytes of UDP-encapsulated data in a user-defined interval of one to 60 seconds. The eight bytes of data consist of the ASCII characters 0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, and 0x38.

You'll need a wireless-capable router and a laptop or desktop PC to run the Phoenix UDP demo. Configure your wireless router to broadcast an SSID of PHOENIX_NETWORK. The Phoenix UDP demo is expecting to join a network with an IP address of 192.168.1.xxx. So, enable your router's DHCP server engine and configure your router to issue IP addresses that begin with 192.168.1.100.

Form your initial network with a single PC, the router, and the WLAN Phoenix. The PC should pick up the first DHCP lease which is 192.168.1.100 and your router should be set to assume the IP address of 192.168.1.1. The WLAN Phoenix does not participate in DHCP IP address assignment. The WLAN Phoenix UDP Demo Utility allows the user to specify the remote PC and local WLAN Phoenix IP addresses and the remote and local UDP ports that the Phoenix UDP demo application should use.

If you're proficient with coding PC

applications, you'll want to whip up a UDP receiver program to run on it. On the other hand, a much easier way to realize a UDP receiver on your computer is to download the free Microchip TCP/IP stack from the Microchip website. Within the files of the stack package, find a file called MCHPDetect.exe which is a Windows application that captures UDP packets sent to UDP port 30303.

Once you have the router, WLAN Phoenix, PC, and software tools in place, start up a terminal emulator on your computer and set the baud rate for 57600 bps. Connect its RS-232 port to the WLAN Phoenix's serial port and apply power to the Phoenix. If you've done your homework, you should see a likeness of Screenshot 2 in your terminal emulator window. At this point, the CompactFlash 802.11b radio card is inactive until you hit the ESC key. While the WLAN Phoenix UDP demo application is in this state, you can enter the desired IP addresses and the UDP port numbers just as I did in Screenshot 2. If you're wondering where the 8088 UDP port number came from, the WLAN Phoenix is hard-coded to receive UDP packets via port 8088 decimal. You can easily change the WLAN Phoenix's hard-coded default port number and default IP address in the WLAN Phoenix firmware driver.

Before we start up the CompactFlash 802.11b radio card, we need to start the MCHPDetect application. We want to be sure to capture any UDP packets that may be thrown about by the WLAN Phoenix. With the MCHPDetect application running in a Windows command prompt window, I activated the CompactFlash 802.11b radio card with a thump of the ESC key. The terminal emulator window refreshed to reveal the hardware (MAC) addresses of the CompactFlash 802.11b radio card plugged into the WLAN Phoenix, as well as the hardware address of my laptop's Ethernet adapter, which is receiving UDP packets from the WLAN Phoenix.

Note the hardware address zeroes in Screenshot 2 have been replaced with a pair of six-byte MAC addresses in Screenshot 3. Checking the validity of the local hardware address is easy as each CompactFlash 802.11b radio card has its unique hardware address printed on its back side. You can verify the remote hardware address by issuing an IPCONFIG /all command on your PC. The IPCONFIG results will also contain the IP address of your computer, which should match the IP address you entered as the remote IP address. Meanwhile, as you can see in Screenshot 4, UDP packets are clocking into the computer at a rate of one per second.

GOT TO PACKET UP AND GO

To make it easy to add the WLAN Phoenix to your Design Cycle, I'm offering the WLAN Phoenix as a full kit and in assembled and tested form on the EDTP Electronics website (**www.edtp.com**). I've provided a mobile networking base that allows you to collect, process, and transfer data over a wireless Ethernet link. What you bounce around your network is up to you.

For those of you that may want to customize the Phoenix firmware driver, it was written using HI-TECH PICC-18 PRO. As always, if you need help getting your WLAN Phoenix online, I'm just an email away. **NV**

SOURCES

Digi-Key
www.digikey.com

EDTP Electronics, Inc.
www.edtp.com
WLAN Phoenix; 802.11b CompactFlash Radio Card

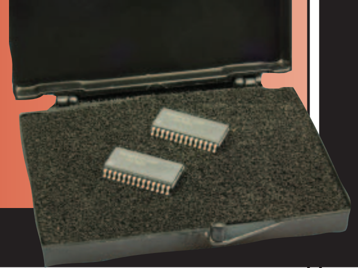
HI-TECH Software
www.htsoft.com
HI-TECH PICC-18 PRO

Jameco
www.jameco.com

Microchip
www.microchip.com
PIC18LF6722; MPLAB IDE; MPLAB REAL ICE

Mouser
www.mouser.com

■ *Fred Eady can be contacted via email at fred@edtp.com.*



■ BY RON HACKETT

BUILD THE PICAXE IR MULTI-BOARD

A Flexible Learning Tool for Infrared Explorations

THIS MONTH, WE'RE GOING TO develop and test our first I/O device for use with the PICAXE-28X1: an 08M-based, multi-function infrared board capable of implementing the entire range of PICAXE IR communications.

Then, in the next column, we'll interface the board with our 28X1 master processor and focus on the software needed to carry out a variety of IR communications tasks. The IR Multi-Board (IRMB) communicates serially with our 28X1 master processor so we will also be exploring the 28X1's new "hserin" command which enables us to automatically receive serial data in the background while our master program is tending to other matters. In addition, the 28X1's new "setintflags" command allows us to trigger an interrupt routine whenever serial data is received, so our master program can immediately respond to the incoming data.

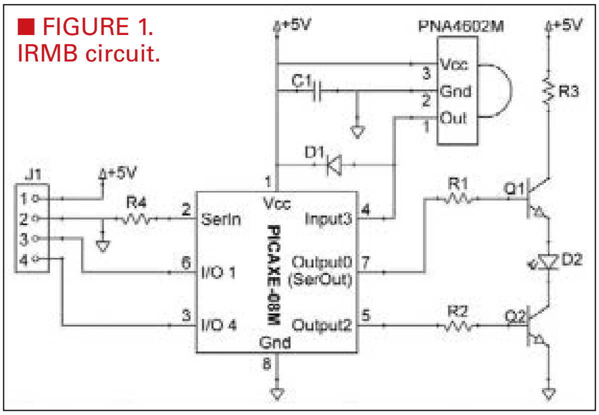
PICAXE IR CAPABILITIES

Current PICAXE controllers support a total of five built-in commands designed to automatically handle the details of the reception and transmission of infrared signals: For IR input, the commands are "infrain" (18X), "infrain2" (18X, 08M, 14M, and 20M), and "irin" (28X1 and 40X1); for IR output, there's "infraout" (08M, 14M, and 20M) and "irout" (28X1 and 40X1). (Note that the 18X does not support IR output.) The three input commands function very similarly as do the two output commands, but there are a couple important differences. The X1 commands (irin and irout) can be used with any input or output pin respectively, whereas the other commands each only function with one specified pin. Secondly, the X1 irin command has a timeout feature that allows your program to move on to other tasks if a valid IR signal is not received within a time period you can specify; infrain and infrain2 pause your program execution indefinitely until a valid IR signal is received.

Since the IRMB uses the 08M processor, we'll focus on the infrain2 and infraout commands (for details of the other IR commands, see Part II of the online PICAXE manual). In order to understand the infrain2 and infraout commands, we first need to take a brief look at the functionality of most modern IR detectors (including the Panasonic PNA4602M that we will be using). In addition to filtering out visible light (which increases reliability), a typical IR detector automatically demodulates a modulated input signal. In other words, the IR input isn't a simple on or off signal; it must be super-imposed on a specific carrier frequency. The PNA4602M expects a 38 kHz carrier wave and the PICAXE infraout command automatically implements the necessary details of signal modulation for successful reception by the PNA4602M. In addition to these built-in IR commands, there are two other methods of implementing IR communications that take an entirely different approach based on the IR detector's 38 kHz carrier wave frequency. All current PICAXE processors support a "pwmout" command that is most commonly used to implement variable speed control for DC motors. Pwmout accomplishes this by generating a continuous pulse-width modulated (PWM) output in the background while your program code executes.

In other words, it produces a rapidly oscillating signal with a specified frequency and duty cycle. For a DC motor, higher duty cycles produce greater speed because the motor is being powered for a greater percent of the time. However, for our current purposes, a PWM signal is essentially a carrier wave, so a 38 kHz PWM IR signal can be easily detected by the PNA4602M. The 4602 is an active-low device which means that its output is normally high and whenever a signal is detected, the output goes low.

The second approach to PICAXE IR communications involves modulating a standard serial output signal with a 38 kHz PWM signal and using it to drive one or more IR LEDs. On the other end of the communications link, a second PICAXE uses an IR detector to demodulate the incoming signal and convert it back to the standard serial protocol. Because the IR detector requires a few oscillations before it actually detects the incoming signal, the serial baud rate is somewhat limited using this approach. However, IR serial communication links using speeds up to 1200 baud can be reliably implemented. The third area of PICAXE IR capabilities also involves the pwmout command, but in an entirely different way. In this case, both the IR transmission and



■ FIGURE 1. IRMB circuit.

IRMB PARTS LIST

■ FIGURE 2

ITEM/DESCRIPTION
- ■ C1/.01 µF bypass capacitor
- ■ D1/1N4148 diode
- ■ D2/Infrared LED
- ■ Heat-shrink tubing to fit IR LED
- ■ IC1/Machine pin IC socket (eight pins)
- ■ J1/Four pin rt. angle female header
- ■ PNA4602/PNA4602M IR detector
- ■ R1, R2/1K (1/4 W) resistor
- ■ R3/47 ohm (1/4 W) resistor
- ■ R4/100K (1/4 W) resistor
- ■ Q1, Q2/KSP2222A NPN transistor

reception are implemented on the same processor, thereby enabling it to function as an object detector for a robot or a motion detector for home control projects. Essentially, a short 38 kHz IR signal is transmitted and the processor immediately listens for an echo. Objects in front of the IR detector within four or five feet can be reliably detected using this approach.

IR MULTI-BOARD FUNCTIONALITY

The IRMB is designed to be a flexible learning tool for the exploration of PICAXE infrared commands and capabilities. It contains all of the circuitry necessary to implement all three IR functions discussed above. The IRMB schematic diagram (Figure 1) is simple but surprisingly powerful and flexible. Essentially, the circuit can be divided into three functional units. First, to the left of the 08M there's a four-pin connector for power and interfacing with our 28X1 master processor. R4 is the 100K resistor we have used previously to tie the 08M's serin pin to ground so that the program will run properly without the external programming adapter (more on that later). The second functional unit (top center) includes the Panasonic PNA4602M IR detector we used back in February of this year. Notice that it's connected to input3, which is the only input on the 08M that is capable of receiving infrain2 commands. Also, as you can see, there is a 1N4148 diode included in this portion of the circuitry. According to the PICAXE documentation, the internal structure of the 08M requires that the diode be included in order for the serin command to function properly. I don't know exactly why this is the case, but I do know that the circuit doesn't work correctly without it because I forgot to include it at first! Also note the diode's orientation; its cathode is connected to +5V (pin 1 of the 08M). The third functional unit (on the right side) is the IR LED driver circuitry. This is what gives the IRMB its power and flexibility; it also requires a little explanation. As we just discussed, there are two very different ways to drive an IR LED. The first uses the SIRC protocol that was introduced last February. Back then, we only received SIRC transmissions from a TV remote control but the 08M is also capable of transmitting SIRC data and the IRMB includes the necessary circuitry. The command for doing so is infraout, which can only be used with the 08M's output0 (pin 7). The second method of IR output that we mentioned involves the 08M's pwmout command, which is only implemented on output2 (pin 5).

The transistors (Q1 and Q2) in Figure 1 serve two important functions. First, the IR LED that I'm using is capable of handling 100 mA, but a PICAXE output pin is limited to 25 mA. The first IRMB prototype that I developed did not use transistor drivers and it had a very disappointing range. For example, it could only respond to an IR remote control within 10 feet.

When I added the transistor drivers, the current through the IR LED increased to 73 mA and the range increased to at least 20 feet (which is as far away as I could get in my basement workspace). In addition, the two transistor drivers function similarly to an AND gate; in order to energize the IR LED, both drivers must be turned on by raising the corresponding 08M output pin. In the next installment of the Primer, we will see how we can use this to our advantage to implement either SIRC IR or serial IR output using the same hardware.

BUILDING THE IRMB

At this point, you could set up a breadboard circuit using the IRMB schematic and the Parts List in Figure 2. You may already have many of the required parts — if not, heat-shrink tubing should be available at any RadioShack store and all of the other parts are available on my website (**www.JRHackett.net**). If you prefer, you can construct a stripboard version of the IRMB. (You knew that was coming — didn't you!) The top and bottom stripboard layout is shown in Figure 3; as usual, the layouts are labeled similarly to a spreadsheet. In the layout and in my stripboard, I used two separate female headers in place of the 08M IC socket so that the wiring underneath the socket would be clearly visible. If you decide to construct the stripboard, it's easier to use an eight pin machine socket as specified in the Parts List. Figure 4 is a close-up of the top and bottom of the completed stripboard. The extra solder slightly below the upper right-hand corner of the bottom view is because I initially soldered the PNA4602M into the board upside down. (Don't you just hate it when you do that!) I also forgot to remove the four-pin male header that I used to insert the IRMB into a breadboard, but it does show you how the connection is accomplished.

Constructing the IRMB stripboard is fairly straightforward, except that it's *critically* important to install four of the parts in the correct order. If you adhere to the following four-step approach, everything should proceed smoothly:

1) With the exception of the parts listed in Steps 2 and 4 that

follow, solder all parts (from small to large) into their appropriate places and trim the leads. When you install the IR-LED, do not insert it as far as it can go. Keep the bottom of the plastic lens about 1/4 inch above the top of the stripboard. This will avoid major problems (unwanted feedback from the LED to the 4602) when we program the IRMB to be an object detector in the next installment.

2) Install the eight-pin socket and by-pass capacitor C1. On the bottom of the board, bend one lead of C1 so that it makes contact with pin 1 of the IC socket, then solder and trim the leads.

3) Using a file or emery paper glued to a flat surface, sand the solder joints on the bottom of the board — especially those that will be underneath the four-pin female right-angle header when it is installed in the next step.

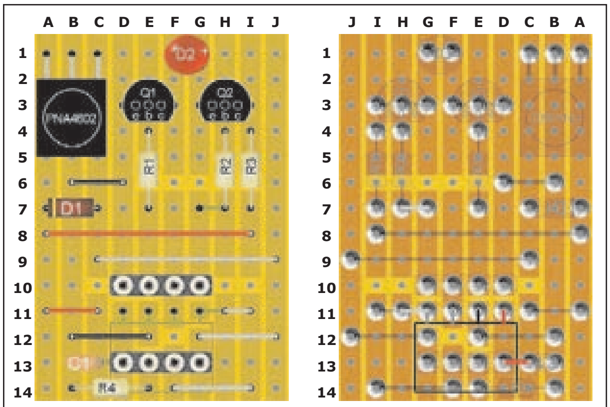
4) Install the four pin female right-angle header on the bottom of the board and the short jumper wire that runs on the top of the board between I11 and H11 and then on the bottom of the board between H11 and G11 (pin 4 of the female header). Hold the header in place with a small spring clamp, making sure that it sits reasonably flat against the bottom of the board and that the jumper wire makes contact with pin 4 of the header at G11. Solder the leads and trim the excess.

When you have completed the IRMB stripboard (or constructed the circuit on a breadboard), we are ready to move on to the software phase of our project.

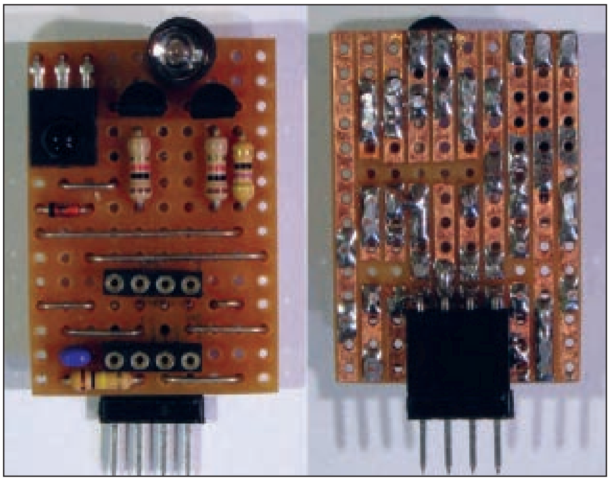
IR MULTI-BOARD SOFTWARE

You can use the IRMB to design, develop, and debug a variety of IR projects. When a project is complete and fully tested, you have two options: use the IRMB itself in the final design or construct a simpler board containing only the necessary circuitry and save the IRMB for the development of your next IR project. For example, if your project only uses IR input capabilities (like the IR-remote input project we are about to discuss), all you need is the PNA4602M — you can leave out the IR LED and transistor driver circuitry. By this time, you're probably wondering how we're going to download a program to the IRMB, since it doesn't have connections for a programming adapter and the Sin and Sout lines aren't brought out to the four-pin connector. The answer is simple: We're going to construct an adapter for the three- or four-pin programming adapter you have already been using! This adapter is so simple that the close-up photo in Figure 5 is all you will need.

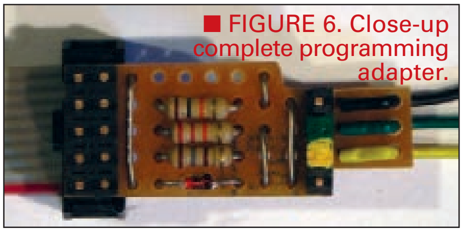
The stripboard I used is a small piece with three traces containing four holes each; no cuts need to be made on the bottom. The board has a three-pin female header soldered to one end of each trace and three color-coded micro-clip leads soldered in parallel with the header pins. The traces on the board actually only need to be two holes long, but I make them longer to strengthen the mechanical connection of each lead. I drilled out the end hole of each four-hole trace so that the wire leads (with insulation) could be threaded up through the end holes, with the bare ends threaded back down and soldered on the bottom as shown. To use the three-pin micro-clip adapter, simply line up the matching colors, plug your three- or four-pin programming adapter into it (as shown in Figure 6), and connect each lead to the appropriate 08M pin (or anywhere else you can access the correct signal). Figure 7 shows the final assembly installed on my 28X1 master circuit ready to be programmed. The first program we'll implement on the IRMB is a simple one that I used to test the functionality of its IR reception circuitry. Similarly to the one we used last February, the program waits for an input from an SIRC TV remote and then displays the received value. However, rather that displaying the value on LEDs as we



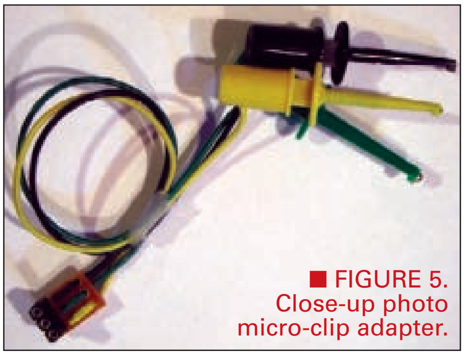
■ FIGURE 3. IRMB top and bottom stripboard layout.



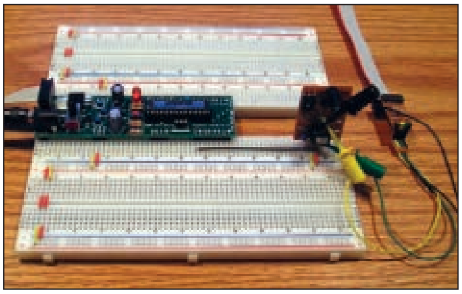
■ FIGURE 4. Top and bottom views of completed stripboard.



■ FIGURE 6. Close-up complete programming adapter.



■ FIGURE 5. Close-up photo micro-clip adapter.



■ FIGURE 7. IRMB programming setup.

did then, this time we'll send the value out to the terminal window of the Programming Editor software.

A couple of the features of the program (see Figure 8) may require explanation. First, the "#terminal 4800" directive may be new to some readers. When the program is downloaded to the PICAXE, it automatically opens a terminal window (at the specified baud rate) within the Programming Editor software. This directive can be used in conjunction with the "sertxd" command, which sends serial data out through the serout programming pin. Since sertxd only functions at 4800 baud, it's necessary to set that rate for the #terminal directive, as well. Using these two features together can be an effective means of debugging programs. Subsequent to the initial download of the program, it may be necessary to manually open the terminal window by choosing "terminal" under the "PICAXE" menu of the Programming Editor.

Second, you probably noticed that the program uses a variable named "infra" without having declared it — this is because infra is already built into the system and there is no need to declare it. Infra automatically contains the most recent SIRC data value received by the IR detector. Infra is actually assigned to variable b13 so whenever you use it in a program, never try to assign another variable name to b13. If you do, it won't be flagged as an error but your program may behave erratically.

Third, the "#" symbol in the argument of the sertxd command is crucial. It formats the output as ASCII decimal characters rather than raw data. In other words, if we had written "infra" rather than "#infra," pressing the Mute button on the remote would cause the program to send the ASCII character 21 to the terminal (remember, we added 1 to the actual infain2 value of 20 for Mute) and ASCII 21 is not a printable character (it would show up as a dark bar in the terminal).

With the "#" included, the ASCII character 21 will be sent to the terminal as the two separate digits "2" and "1" — try the program with and without the "#" to see what happens. Incidentally, 13 and 10 are the ASCII characters for carriage return and line feed; they just format the output so that each TV remote key-press is printed on a new line. If you need a copy of the ASCII chart, just search for it online — there are thousands of links.

Finally, the program's use of the "do... loop" command may also require clarification. Do... loop is a very powerful structure that can be used to carry out a variety of tasks. In its simplest form (i.e., without an "until" or "while" qualifier), it simply creates an infinite loop, so in this case it serves the same purpose as the familiar "main: ... goto main" structure that we have used in the past to loop "forever."

```
' == IRMB_Remote.bas ========== 2008.07.26 ========= Ron Hackett ==

'     This program runs on a PICAXE-08M processor at 4 MHz.
'     It waits for a key-press from an SIRC TV remote control,
'     encodes it & transmits it (as ASCII digits) to the
'     Programming Editor's Terminal Window.

' == Directives ==

      #com1                      ' specify com port for downloads
      #picaxe 08M                ' specify PICAXE processor
      #terminal 4800             ' open terminal window at 4800 baud


' == Begin Main Program ==========================================

'     Note: "infra" is a built-in variable (actually b13)

      do                         ' The following is an infinite loop:
        infrain2                 '  wait here for infra-red input
        infra = infra + 1        '  add 1 so digits & data correspond
        sertxd (#infra, 13, 10)  '  send ASCII digits to the Terminal
                     pause 250   '  slow it down a bit
      loop                       '  loop forever
```

■ FIGURE 8. IRMB TV remote input program.

The program is simple enough to type directly into the Programming Editor, so give it a try and take your IRMB out for a spin. If at first it doesn't function as expected, you'll need to go through the usual debugging procedures: Check your board for unwanted solder-bridges or other wiring problems (I had an unwanted solder bridge that crashed my IRMB the first time out); try a simpler program (e.g., just create a simple loop to repetitively send the digits 0 through 9 to the Terminal Window). As usual, persistence (and a logical debugging strategy) is the key.

TO BE CONTINUED ...

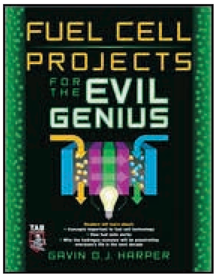
In the next installment of the Primer, remember we'll get into the details of interfacing the IRMB with our 28X1 master processor. We'll also experiment with an IR communication link that will require two IRMB circuits (or portions thereof), so you might want to construct a second one after you get the first one up and running. Alternatively, you can use a simple breadboard circuit for one end of the link (battery power would be helpful if you want to do any range testing) or you might be interested in the printed circuit board (PCB) version of the IRMB. A prototype is shown in Figure 9 and the final production version should be available on my website by the time you are reading this. As you can see, in Figure 9, the PCB version does include a three-pin female header to which you can attach your programming adapter, so it doesn't require the additional multi-clip one. In the meantime, you might want to take a look at Part II in the manual for the 28X1's "hserin" and "setintflags" commands. The manual's discussion of the new "scratchpad" memory would also be helpful. If you like a challenge, you might even try to interface your IRMB to the 28X1 before we actually do it here. Whatever you do, have fun! NV

■ FIGURE 9. IRMB PCB prototype.

We have over 170 related titles in our webstore!!

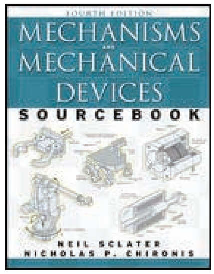
The Nuts & Volts WEBSTORE

"EDITOR'S PICKS"



Fuel Cell Projects for the Evil Genius

Interest in fuel cell technology is on the rise, thanks to the new focus on environmental awareness. This excellent addition to the Evil Genius series continues the tradition of presenting practical examples of technologies you can build and learn from.
$24.95



Mechanisms and Mechanical Devices Sourcebook, Fourth Edition

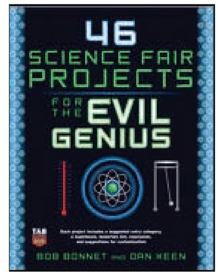
If you're serious about designing and building your own robot, you shouldn't be without this book.
Each of the over 2,000 illustrations is worth at least 1,024 words.
$89.95

Editor Bryan Bergeron's recommended reads.
Find these and many more great titles in the *NUTS & VOLTS* Webstore!

NEW!

46 Science Fair Projects for the Evil Genius
by Bob Bonnet and Dan Keen
Pub. Date 08/06/08

SHAKE UP YOUR SCIENCE FAIR WITH THESE CUTTING-EDGE, ATTENTION-GRABBING PROJECTS!

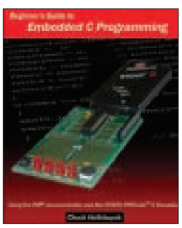


Want your kids to win first place in the next science fair? *46 Science Fair Projects for the Evil Genius* has everything you need to create amazing, sophisticated projects that will wow the judges and keep everyone talking long after the awards are handed out.
$19.95

ELECTRONICS

Best Seller

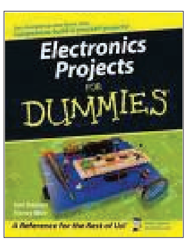
Beginner's Guide to Embedding C Programs
by Chuck Hellebuyck



The C language has been covered in many books but none as dedicated to the embedded microcontroller beginner. Through his down-to-earth style of writing, Chuck Hellebuyck delivers a step-by-step introduction to learning how to program microcontrollers with the C language. In addition, he uses a powerful C compiler that the reader can download for free in a series of hands-on projects with sample code so you can learn right along with him. For the hardware, he found the best low cost but effective development starter kit that includes a PIC16F690 microcontroller and everything else the beginner needs to program and develop embedded designs.
$39.95

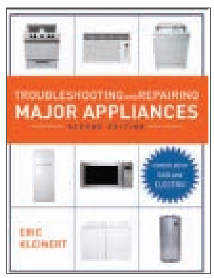
Electronics Projects For Dummies
by Earl Boysen & Nancy C. Muir

NEW!



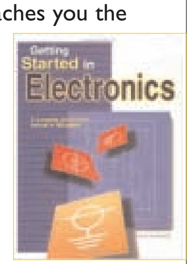
Who says the Science Fair has to end? If you love building gadgets, this book belongs on your radar. Here are complete directions for building 10 cool creations that involve light, sound, or vibrations — a weird microphone, remote control gizmos, talking toys, and more, with full parts and tools lists, safety guidelines, and wiring schematics.
$24.95

Troubleshooting and Repairing Major Appliances: 2nd Edition
by Eric Kleinert



Now covering both gas and electric appliances, the updated second edition of *Troubleshooting and Repairing Major Appliances* offers you a complete guide to the latest tools, techniques, and parts for troubleshooting and repairing any appliance. Packed with over 200 illustrations, the book includes step-by-step procedures for testing and replacing parts, instructions for reading wiring diagrams, charts with troubleshooting solutions, advice on using tools and test meters, safety techniques, and more. **$59.95***

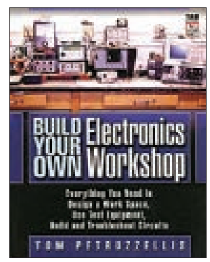
Getting Started in Electronics
by Forrest M. Mims III



Author Forrest Mims teaches you the basics, takes you on a tour of analog and digital components, explains how they work, and shows you how they are combined for various applications. Includes circuit assembly tips and 100 electronic circuits and projects you can build and test. Forrest M. Mims, III, has written dozens of books, hundreds of articles, invented scientific devices, and loves to share his knowledge with eager students! **$19.95**

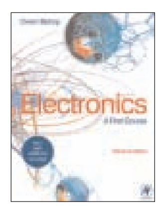
Build Your Own Electronics Workshop
By Thomas Petruzzellis

YOUR DREAM ELECTRONICS LAB IS WAITING INSIDE!



This value-packed resource provides everything needed to put together a fully functioning home electronics workshop! From finding space to stocking it with components to putting the shop into action — building, testing, and troubleshooting systems — popular electronics author Tom Petruzzellis' *Build Your Own Electronics Workshop* has it all! And the best part is, this book will save you money, big time! Reg **$29.95** **Sale Price $24.95**

Electronics: A First Course Second Edition
by Owen Bishop

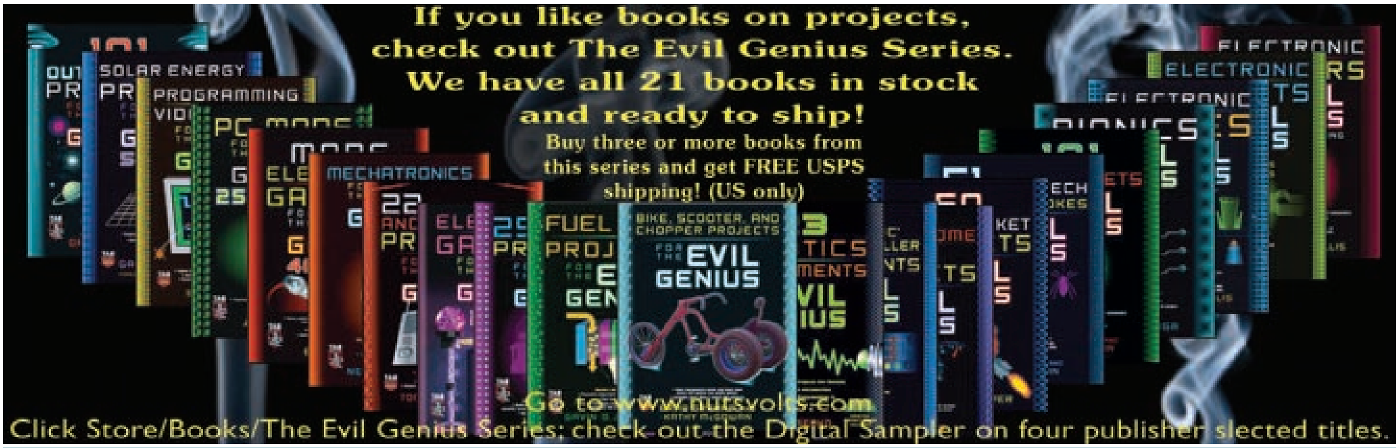


Electronics: First Course starts with the basics of electricity and component types, and introduces students to practical work almost straightaway. No prior knowledge of electronics is assumed. The second edition now has additional applicability to BTEC First Electronics from Edexcel with coverage of fundamental topics required by students of this qualification, as well as other essential new topics that reflect recent technological developments.**$36.95**

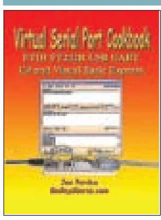
Order online @ www.nutsvolts.com

Or CALL 1-800-783-4624 today!

THE EVIL GENIUS SERIES



BOOK & KIT COMBOS



Virtual Serial Port Cookbook
by Joe Pardue
As talked about in the *Nuts & Volts* June issue, **"Long Live The Serial Port"**

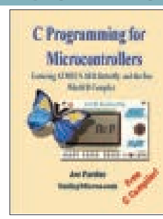
Kit **$69.95**

Book **$44.95**

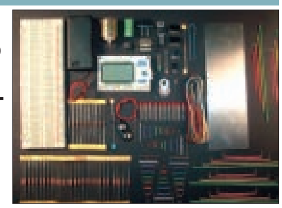
This is a cookbook for communicating between a PC and a microcontroller using the FTDI FT232R USB UART IC. The book has lots of software and hardware examples. The code is in C# and Visual Basic Express allowing you to build graphical user interfaces and add serial port functions to create communications programs.

The Virtual Serial Port Parts Kit and CD
(also *available, above right*)

Combo Price $115.95 Plus S/H $8.95



From the Smiley's Workshop
page 55
C Programming for Microcontrollers
by Joe Pardue



Book **$44.95**

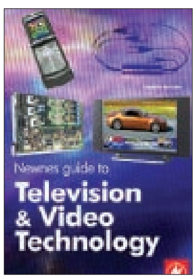
Kit **$66.95**

Do you want a low cost way to learn C programming for microcontrollers? This 300 page book and software CD show you how to use ATMEL's AVR Butterfly board (*available in the kit, above right*) and the FREE WinAVR C compiler to make a very inexpensive system for using C to develop microcontroller projects.

Combo Price $99.95 Plus S/H $8.95

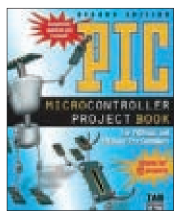
Best Seller

Newnes Guide to Television and Video Technology
by K. F. Ibrahim



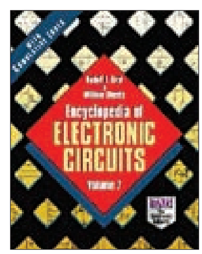
Starting with TV fundamentals, the bulk of the book covers the many new technologies that are bringing rowth to the TV and video market, such as plasma and LCD, DLP (digital light processing), DVD, Blu-ray technology, Digital television, High Definition television (HDTV), and video projection systems. **$34.95**

PIC Microcontroller Project Book
by John Iovine



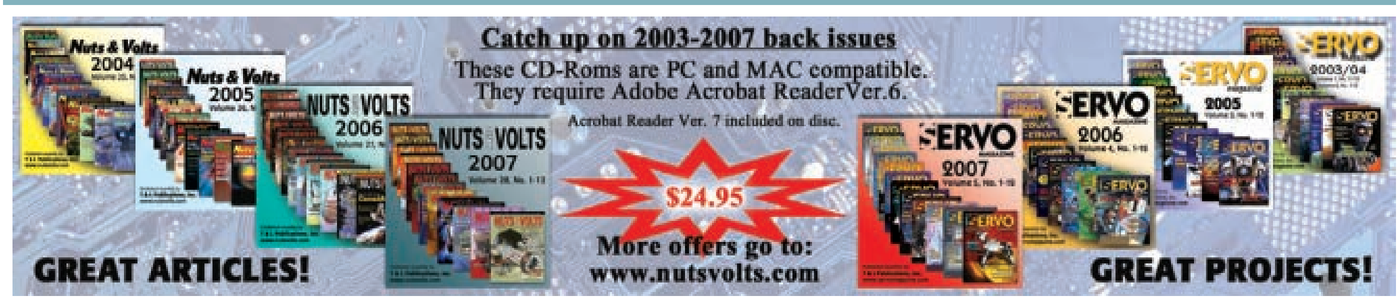
The PIC microcontroller is enormously popular both in the US and abroad. The electronics hobbyist market has become more sophisticated. This new edition is fully updated and revised to include detailed directions on using both versions of the microcontroller, with no-nonsense recommendations on which is better served in different situations. **$29.95**

Encyclopedia of Electronic Circuits, Volume 7
by Rudolf F. Graf / William Sheets
Publish Date: August 31, 1998



An extensive library of 1,000 circuits from the bestselling, seven-volume *Encyclopedia of Electronic Circuits*. A virtual treasurehouse. An invaluable reference tool for every hobbyist, technician, student, and design professional. **$39.95**

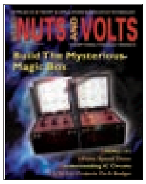
CD-ROMS



CALL 1-800-783-4624 today!
Or Order online @ www.nutsvolts.com

PROJECTS

Magic Box Kit



As seen on the April 2007 cover

SALE!

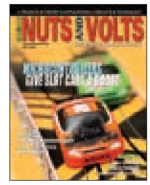
This project blends electronics technology with hand craftsmanship. The clever trick has the observer remove one or two of six pawns while you are out of the room and upon re-entering you indicate the missing pawns without ever opening the box.

For more info go to our webstore @ www.nutsvolts.com

Reg. Price **$47.55** Sale Price **$37.95**

Freeze Fountain Kit

As seen in the May 2008 issue



Looking for a fun project that has a great visual effect, then check out The Freeze Fountain PCB & Component Kit.

This kit has all the components you need to start amazing your friends by creating the illusion of water drops *frozen* in time. You can even show them dripping upwards!

Sold as a kit, or the PCBs can be puchased separately.

For more information, please check out your May 2008 issue or go to our webstore @ www.nutsvolts.com

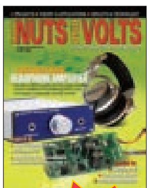
KIT Subscriber's Price **$98.95** Non-Subscriber's Price **$109.55**

PCBs Subscriber's Price **$34.95** Non-Subscriber's Price **$39.95**

New optional hardware avalible!

Switching Regulator Kit

Need to regulate your power output? Take a look at this new and improved regulator. **For more information, please check out your June 2008 issue or go to our webstore @ www.nutsvolts.com**



Best Seller

Subscriber's Price **$29.95**

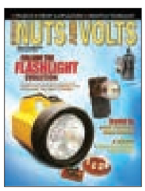
Non-Subscriber's Price **$32.95**

Kit includes an article reprint.

PCBs can be bought separately.

Digital LED Conversion Kit

200% More Efficient



As seen on the February 2008 cover

Turn all your old style bulb flashlights into bright, five LED, energy efficient, 4.5 VDC digital flashlights.

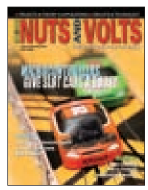
For more information, go to our website @ www.nutsvolts.com

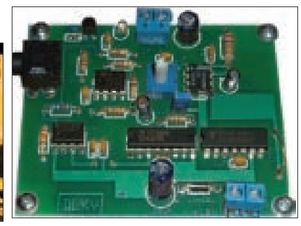
Without Flashlight Housing **$22.49**

With Flashlight Housing **$25.95**

Both include an article reprint.

Voice Changer Kit





Have fun with the whole family with this project. Put your newly ***changed*** robotic ***voice*** on your outgoing phone message, or use it on those annoying telemarketers!

For more information, please check out your May 2008 issue

Subscriber's Price **$39.95**

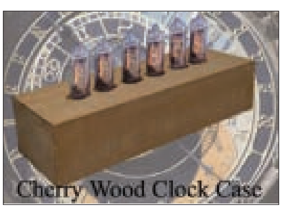
Non-Subscriber's Price **$45.95**

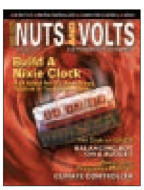
PCBs can be bought separately.

$ 20.95*

Both include an article reprint.

Nixie Clock Kit





October 2006

Nixie tube clocks fuse the spirit, drama, and eerie beauty of cold war technology with modern inner works to create uncommon handcrafted timepieces.

Now with optional case choices!

Get more info @ www.nutsvolts.com

Subscriber's Price **$146.95**

Non-Subscriber's Price **$159.95**

Kit includes article reprint, complete instructions and parts list.

Talking Skull Kit

NEW!

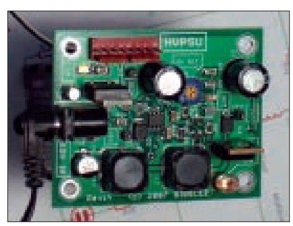
Get ready for a tech-scary Halloween!

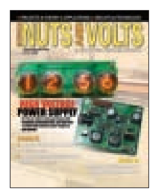
For more information, please check out the September issue or go to our webstore @ www.nutsvolts.com

Subscriber's Price **$69.95**

Non-Subscriber's Price **$73.75**

H.V.P.S. Kit





As seen on the April 2008 cover

It's fun to collect and experiment with forgotten technology! But, you'll need a stable high voltage power supply to get started.

For more info, go to the *Nuts & Volts* Webstore **www.nutsvolts.com**

KIT Subscriber's Price **$49.45**

Non-Subscriber's Price **$54.95**

Both include an article reprint.

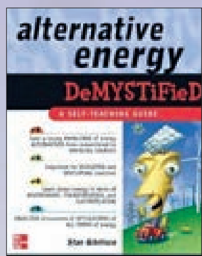
WE ACCEPT VISA, MC, AMEX, and DISCOVER. Prices do not include shipping and may be subject to change.

Attention **Subscribers ask about your discount on prices marked with an ***

The Nuts & Volts WEBSTORE

ALTERNATIVE ENERGY SECTION

Alternative Energy Demystified
by Stan Gibilisco
Publish Date: October 23, 2006



The fast and easy way to get up-to-speed on alternative energy. Because of current events, geopolitics, and natural disasters, the cost of fuel is front and center in our lives. This book provides a concise look at all forms of energy, including fossil fuels, electric, solar, biodiesel, nuclear, hydroelectric, wind, and renewable fuel cells. You will get explanations, definitions, and analysis of each alternative energy source from a technological point-of-view. **$19.95**

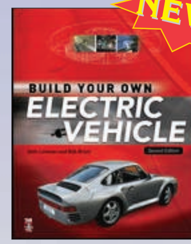
Solar Hydrogen Power Generation Set

The Solar Hydrogen Generation Set includes a solar cell, a PEM reversible fuel cell, as well as oxygen and hydrogen gas containers. The set only needs pure water to create hydrogen and produce electricity. Perfect for science labs, classroom use, or demonstration purposes.
$67.00

Build Your Own Electric Vehicle
by Seth Leitman, Bob Brant
Publish Date: October 10, 2008

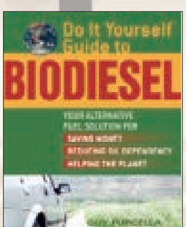
Go Green-Go Electric! Faster, Cheaper, More Reliable While Saving Energy and the Environment!

NEW!



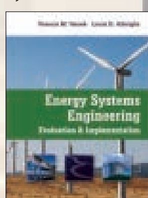
This comprehensive how-to goes through the process of transforming an internal combustion engine vehicle to electric or even building an EV from scratch for as much or even cheaper than purchasing a traditional car. The book describes each component in detail---motor, battery, controller, charger, and chassis---and provides step-by-step instructions on how to put them all together. **$29.95**

Do It Yourself Guide to Biodiesel
by Guy Purcella



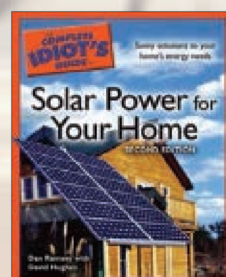
Contains the most current and complete information available for would-be biodiesel makers. This book offers a step-by-step path from the initial desire to do something good for the environment to the final stage of filling the tank with low-cost fuel. Readers are given a complete understanding of what biodiesel is and how to make it properly and safely. There is even a section on pitfalls to avoid, as well as the author's personal tips and tricks. **$15.95**

Energy Systems Engineering
by Francis Vanek, Louis D. Albright
Publish Date: June 13, 2008



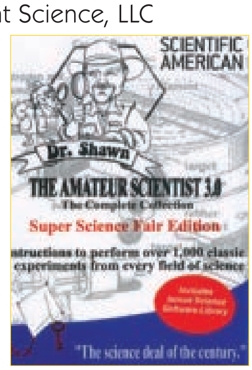
A unique systems approach to energy engineering, covering carbon-based, nuclear, and renewable sources!
Filled with over 200 detailed illustrations and tables, the book examines short-, medium-, and long-term energy options for the remainder of the twenty-first century. For each energy system, the authors provide equations and problems to help practitioners quantify the performance of the technology and better understand its potential. **$89.95***

The Complete Idiot's Guide to Solar Power for Your Home
by Dan Ramsey / David Hughes
Publish Date: May 2007



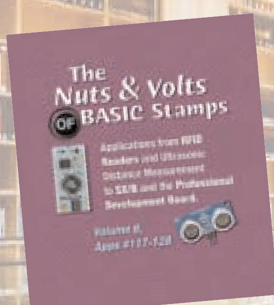
The perfect source for solar power — fully illustrated. This book helps readers understand the basics of solar power and other renewable energy sources, explore whether solar power makes sense for them, what their options are, and what's involved with installing various on- and off-grid systems. **$19.95**

The Amateur Scientist 3.0 The Complete Collection
by Bright Science, LLC



There are 1,000 projects on this **CD**, not to mention the additional technical info and bonus features. It doesn't matter if you're a complete novice looking to do their first science fair project or a super tech-head gadget freak; there are enough projects on the single CD-ROM to keep you and 50 of your friends busy for a lifetime! **$26.99**

Special Offer



FREE! with any purchase over $100
Phone-in orders only!

FREE
***Nuts & Volts* BASIC Stamp Vol 6 with orders over $100.00 of any product combination!**
Offer ends October 31st, 2008.
Call 1 800 783-4624

CLASSIFIEDS

COMPONENTS

RF PARTS™ CO.
Tubes, Transistors, Power Components
Email: rfp@rfparts.com • Web: www.rfparts.com
800-737-2787 Fax 888-744-1943

STA PARTS ELECTRONIC COMPONENTS
Obsolete & current production parts are out there if you know where to find them. STA specializes in finding these hard to find parts for you. Call today and end your search with STA.
Personalized Service — No high pressure salesmen
www.staparts.net 888-869-6999

uM-FPU V3.1
Floating Point Coprocessor

32-bit IEEE 754
SPI or I2C
DIP-18, SOIC-18
Fast – Easy to Use
Extensive floating point support, GPS input, FFT, 12-bit A/D, matrix operations, user-defined functions.
www.micromegacorp.com

LEDs
Super Brights from 20¢
RGB Color Mixing 30 Pack $6.99
Assortment 120 Pack $24.99
Shipping from $3.00
Alan-Parekh.com/Store

Easy to use, efficient
Voltage Regulators!

- Integrated capacitors (no external components)
- Switching regulators give 80+% efficiency
- Drop-in replacement for LM78xx
- Step down, step up, fixed and adjustable versions

Starting at $15
dimensionengineering.com

SOLAR PRODUCTS/KITS

BATTERY DESULFATORS
www.wizbangplus.com

Micro Circuit Labs
ELECTRONIC KITS
www.microcircuitlabs.com

Solar Data Logger
- Logs irradiance for 170 days
- Reports kWh/m²
- Weatherproof enclosure

SDL-1 Kit $69.95
Solar Blinker – SB1 Kit $19.95
Solar Irradiance Meter – SIM Kit $22.95
Adjustable Voltage Regulator – AVR - 1 Kit $14.95

VINTAGE ELECTRONICS WANTED

WANTED
60s-70s Wang calculators & computers, 1970s Intel computers, development systems, peripherals & all related Intel & Wang literature.
Randy, 1309 Gatesmeadow Way, Reston, VA 20194-1426, 703-447-4223, apergy@aol.com

WANTED
1960s Mathatron calculators by Mathatronics and 1960s Wyle Laboratories calculators and other unusual electronic calculators and all related literature.
Randy, 1309 Gatesmeadow Way, Reston, VA 20194-1426, 703-447-4223, apergy@aol.com

ROBOTICS

$29.95 MaxSonar-EZ1
(MSRP)

High Performance Ultrasonic Range Finder
- serial, analog voltage & pulse width outputs
- lowest power - 2mA
- narrow beam
- very easy to use!

www.maxbotix.com

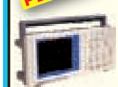
ARobot Kit from Arrick Robotics uses the Basic Stamp II. Quality metal construction. Easy to assemble and very expandable.
$235.00
www.robotics.com/arobot

HARDWARE WANTED

DEC EQUIPMENT WANTED!!!
Digital Equipment Corp. and compatibles.
Buy - Sell - Trade
CALL KEYWAYS 937-847-2300
or email buyer@keyways.com

OSCILLOSCOPES

Bargain Test Gear!


PDS5022
Portable, benchtop 25Mhz 100MS/s 2-ch color LCD Scope. Download waveforms via USB1.1 to PC. 7.8" 640 x 480 Color LCD 3.5lbs. Volt/Time Automatic Measurement Peak-to-peak Value, Average, RMS, Frequency and Cycle. FREE 1x/10x probes, USB cable. Compare with expensive LCD scopes!
PDS5022S $325


SPI Xpress
SPI Bus Analyzer - SPI protocol exerciser and analyzer for standard SPI and non-standard 4-wire and 3-wire serial protocol interfaces up to 50 Mbps. PC control via free GUI Control Panel Software. SPI sniffer, chip-to-chip emulation.
SPI Xpress $783

Saelig.com unique electronics
1-888-7SAELIG
info@saelig.com
www.saelig.com

BOOKS

Books
PIC® Microcontroller
sq-1.com
Stepper Motors
stepperstuff.com
CNC
cncintro.com
SQUARE 1 ELECTRONICS
(208) 664-4115
Hayden, ID USA

HOBBYIST BOOKS
FROM $2.95
"Will fascinate, captivate"
A.R. Winstanley, Electronics Editor
http://stores.lulu.com/thomasscarborough

CONNECTORS/WIRE/CABLE

The RF Connection
213 N. Frederick Ave., Ste. 11NV
Gaithersburg, MD USA 20877
http://www.therfc.com/

Complete Selection of MIL-Spec Coax, RF Connectors and Relays

UG-21B/U N Male for RG-213/214 .$5.00
UG-21D/U N Male for RG-213/214 .$3.25
N Connectors for 9913/Flexi4XL/9096
UG-21B/9913 .....$6.00 / Pins Only......$1.50
UG-21D/9913 ..$4.00 / Extra Gasket.$0.75
UG-176/U Reducer RG-59/8X, $0.25 or 5/$1.00
UG-175/U Reducer RG-58/58A, $0.25 or 5/$1.00
Silver Teflon PL-259/Gold Pin, $1.00 or 10/$9.00

Call for Specials of the Month
Full Line of Audio Connectors for Icom, Kenwood, and Yaesu

8 Pin Mike Female	$2.50
8 Pin Mike Male Panel	$2.50
13 Pin DIN for Kenwood	$2.75
8 Pin DIN for Icom	$1.00
8 Pin DIN for Kenwood	$1.50

Prices Do Not Include Shipping
Prices subject to change without notice.

Orders	800-783-2666
Info	301-840-5477
FAX	301-869-3680

SCSI
I-U320 50Pin-68Pin-80Pin
1 to 8 Bay Case Enclosures
Adapters Cables Terminators
Low Prices - Qty Discounts!
(Also FireWire, USB, Video)
www.mcpb.com

LCDs/DISPLAYS


As low as $10.00ea
- White text on blue background 20x4 LCD
- High contrast & wide viewing angles
- HD44780 equivalent controller built-in
- Bright white LED back light

Order online at: www.411techsystems.com

MISCELLANEOUS FOR SALE

LIQUIDATION SALE
Over 20,000 speakers, $100,000 of speaker parts, glue, speaker boxes and grilles, crossovers, induction heaters, magnitizers, spot welders, chart recorders, transformers, variacs, industrial ovens, motors, electronic equipment. We get rid of what you don't want!
New Address: ISE, 15960 FM-800, San Benito TX, 78586
www.isesurplus.com or 888-351-5550 (toll free)

GADGETS

GREEN LASER POINTER

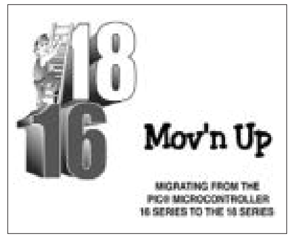
.93 mile range
$22.89 Free Shipping
www.gadgetcentral.ca

CLASSIFIEDS

NEW PRODUCTS

continued from page 26

MOV'N UP



Mov'n Up — a new book from Square 1 Electronics — shows assembly language programmers how to easily migrate from Microchip's 16 series eight-bit microcontrollers to the 18 series devices. The 18 series devices have some features that make them easier to use. Program memory paging is gone, so tables may be of any length and may be located anywhere. Data memory bank selection is simplified. Compare, bit toggle, and set file instructions make writing programs easier. Context saving on interrupt is automatic (sometimes). The newer application peripherals (CAN bus, etc.) are included in the 18 series devices.

Mov'n Up will save readers time by providing an explanation of the fundamental differences (vs. 16 series) along with programming examples to make the transition easier. *Mov'n Up* is available from the publisher ($24.95 plus s&h).

For more information, contact:
Square 1 Electronics
Web: **www.sq-1.com**

TECH FORUM

This is a READER-TO-READER Column.

All questions *AND* answers are submitted by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving problems of a technical nature. Questions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted by readers and NO GUARANTEES WHATSOEVER are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals. Always use common sense and good judgement!

All questions and answers should be sent by email to forum@nutsvolts.com All *diagrams should be computer generated and sent with your submission as an attachment.*

QUESTIONS

To be considered, all questions should relate to one or more of the following:

1. Circuit Design
2. Electronic Theory
3. Problem Solving
4. Other Similar Topics

- Be brief but include all pertinent information. If no one knows what you're asking, you won't get any response (and we probably won't print it either).
- Include your Name, Address, Phone Number, and email. Only your Name, City, and State will be published with the question, but we may need to contact you.
- No questions will be accepted that offer equipment for sale or equipment wanted to buy.
- Selected questions will be printed one time on a space available basis.
- Questions are subject to editing.

ANSWERS

- Include in the subject line of your email, the question number that appears directly below the question you are responding to.
- Payment of $25.00 will be sent if your answer is printed. Be sure to include your mailing address or we cannot send payment.
- Only your Name, City, and State, will be printed, unless you say otherwise. If you want your email address included, indicate to that effect.
- Comments regarding answers printed in this column may be printed in the Reader Feedback section if space allows.

>>> QUESTIONS

I am looking for an efficient, physically very small, low parts count circuit to provide a regulated 5V at up to 10 mA output directly from a power line (115 VAC) input.

#10081 **Bryan**
Houghton, MI

I need to monitor when my sump pump in the basement goes on and off. The pump draws 10A @ l20 VAC. Any ideas?

#10082 **Paul Vandervort**
via email

I have an X10 lighting system in my home that used to work flawlessly, using X10's "Active Home" software to control the yard lights. Over the last year, some zones work intermittently and some not at all. I've tried to find any electrically noisy appliances that may be on when the system gets flakey, but have not seen any consistent pattern that might be causing the problem. I do know that the new pump motor for the pool equipment creates problems when it's on, but it shuts off hours before the lights come on (or try to). Sometimes the AH software can't turn on the lights, but I can do it manually with the remote.

#10083 **Larry**
Corona, CA

>>> ANSWERS

[#8081 - August 2008]

I'm looking for information on power line communication systems. Does the US power grid broadcast any information that can be read from the power lines coming into my home? Are there any projects that can tap that information?

#1 Most distribution power lines have 60 Hz on them and that is it. If you happen to be on a power line with what is called the "Turtle" then there will be data. I do not know the protocol for this data. You don't really want to know what your neighbor's power usage is.

Sometimes there will be a Power Line Carrier. This will carry data, voice, and special signaling, like high speed transfer trip. The problem there is it is blocked from getting to your house by band stop filters on the power line. The filters are not to keep the signal away from you so much as to remove echos from beyond the power line receiver.

Alonzo E. Fuller
Sweet Home, OR

#2 Aclara (formerly Distribution Control Systems) sells the TWACS (Two-Way Automatic Communications System) technology to utilities which allows the utility to read your meter

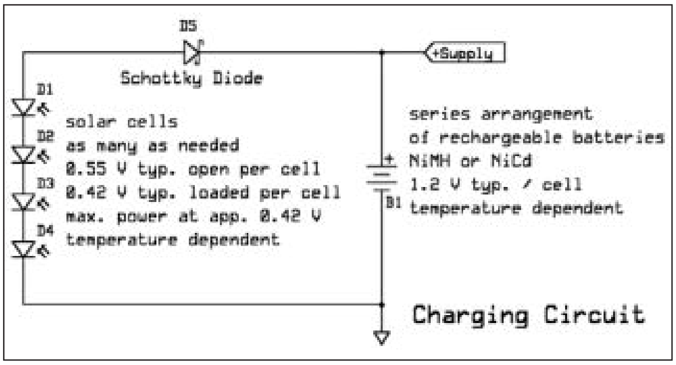
over the power line. They can also control loads, turn your electric off and on, and determine precise power outage maps over the power line. You can read more about TWACS on their website at **www.aclaratech.com**.

It uses a zero-crossing modulation scheme and is neither fast nor efficient. In fact, remember back in the days of the Commodore 64 when you could read a text file as it downloaded at 1200 bps? Yeah, that is lightning speed compared to power line communication.

Jeff Blevins
Saint Louis, MO

[#8082 - August 2008]

I need info on refurbishing or building custom solar yard lights. I currently have a Lighthouse lawn ornament that has a solar light that doesn't recharge the replaceable rechargeable battery. Information on troubleshooting would be helpful for what I assume to be a fairly common circut. I would also like to illuminate the lantern of a Lawn Jockey and a Pagoda. Adding multiple LEDs with changing colors and blinking patterns could also be a take off.



Shown is a suitable charging circuit which consists of a string of solar cells producing a voltage high enough to charge the NiMH or NiCd batteries through a Schottky diode. The diode prevents discharge through the solar cells with insufficient sunlight. In our latitudes, significant current is produced from about 9 am to 3 pm and the solar cells should be sized to produce about 0.2-0.25 It (cell capacity in mAhr @ nominal conditions). Further information can be found in rechargeable battery literature; SANYO has probably one of the best explanations up on their website.

A circuit for a pattern generator is also shown, consisting of an oscillator driving a four-bit up counter connected to the lower addresses of a 2K x 8 EEPROM. This circuit simply reads consecutively the contents of the EEPROM, and you can either store the desired patterns or use an already programmed (recycled) one; you may be surprised by the patterns you get. Other EEPROMS can also be used at either more or less patterns depending upon the size of the EEPROM. The 2K x 8 has space for 64 different patterns, which are adjusted through SW1 through SW6.

Walter Heissenberger
Hancock, NH

[#9083 - September 2008]

I have several Maytag Neptune washing machine motors with the driver board.

I would like to make an adjustable speed generator so that I could use these motors on wood working tools. I was thinking that a 555 series chip would be the basis, but I don't know how to get the three phase output. Also, would I drive one or two phases of the motor at the same time?

One thought was to use the 555 to cut the power to one winding at a time sequentially and leave power on the other phases. I don't know how Maytag did this. I don't want to use a microprocessor chip if I can avoid it.

#1 Appliance manufacturers have switched in the last few years to three phase motors connected to a variable frequency drive, especially on applications requiring several speeds, such as washing machines. It also frequently includes power factor correction ahead of the DC intermediate loop to present an almost purely "resistive load" to the power grid in order to get high efficiency. For this particular application, Microchip Technology makes several boards using either their digital signal processors or their microcontrollers with power modules (three half bridges) from International Rectifier. This will give you some insights and the software is completely open. You can also hack the VFD board; usually there is a connector to a controller. As an alternate, you can use three phase motors connected through a few capacitors to a single phase system at higher efficiency than even connected to a three phase system. There is an excellent report by Dr. Otto Smith, *High Efficiency Air-Conditioner on Single-Phase Electricity*, Energy Innovations Small Grant, State of California (in public domain, on the Internet), revealing some interesting results. This is, however, a single speed application and you may use variable pulleys to change the speed.

Walter Heissenberger
Hancock, NH

#2 The key to success here would be a working schematic for the Maytag designed controller; without it you are guessing in the dark. The board is readily available as a spare part (for under $200), but the schematic was not found by Google search and may be proprietary. The hook-up diagram (found inside the washer rear door) does not specify the voltages and signals needed to operate the motor controller module and doesn't help here.

It would be worth "reverse-engineering" a motor control module and using a DMM and oscilloscope to examine the signals on a working appliance before making any homebrew modifications. At this point, it is not possible to understand how the motor operates, as it could be a true stepper motor with multiple phase windings or a modified induction motor that only requires phase angle control using the AC power line frequency and one or two triac switches.

Frequent mention of replacing over-stressed components on that module was found in the on-line forums for appliance repair. Perhaps that Maytag repairman has less free time on his hands than in the old days!

Peter Stonard
Campbell, CA

AD INDEX

■ LOOK FOR ■ SEARCH FOR ■ FIND

Find your favorite advertisers here!

NEW Single-wire EEPROM Family In a Tiny 3-lead Package!

Microchip Technology's new UNI/O serial EEPROM uses only ONE connection to the host microcontroller. This compares to two or three pins for I²C™ and three to six pins for Microwire or SPI buses. This new, flexible bus offers advanced features like a status register and write protection on demand, along with all I/O, data and command functions through a single pin.

Simplify designs & reduce system cost

- Free up pins on the MCU
  - Enhance your design by adding new features
  - Move to a smaller MCU = lower cost
- Free up pins on your connector
  - Smaller connector = lower cost
- Compact: Tiny packages and no pullup resistors

Optimized for embedded applications

- Software Write Protection ¼, ½, or full array
- Flexible data rate 10 - 100 kHz set by host
- 1-million E/W cycles, 200-year data retention
- Low standby current - 1 µA
- Real EEPROM, up to 125°C

Device	Density (bits)	Operating Voltage
11AA010	1K	1.8-5.5V
11LC010	1K	2.5-5.5V
11AA020	2K	1.8-5.5V
11LC020	2K	2.5-5.5V
11AA040	4K	1.8-5.5V
11LC040	4K	2.5-5.5V
11AA080	8K	1.8-5.5V
11LC080	8K	2.5-5.5V
11AA160	16K	1.8-5.5V
11LC160	16K	2.5-5.5V

GET STARTED TODAY!

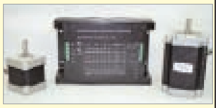
Purchase the **MPLAB® Starter Kit for Serial Memory Products** (DV243003), which supports the UNI/O family of serial EEPROM and all other Microchip serial EEPROMs for $79.98 at www.microchipDIRECT.com.

www.microchip.com/UNIO

MICROCHIP

CircuitSpecialists.com
1000's of Items Online!
1-800-528-1417
Fax: 480-464-5824
Since 1971

Stepper Motors & Stepper Motor Controllers



Part #:	Frame Size:	Holding Torque:	Price:
42BYGH404	NEMA 17	3.4kg.cm/47oz.in	$16.00
57BYGH207	NEMA 23	8kg.cm/111oz.in	$21.50
57BYGH303	NEMA 23	15kg.cm/208oz.in	$25.95
57BYGH405	NEMA 23	20kg.cm/277oz.in	$28.90
85BYGH450B-03	NEMA 34	48kg.in/665oz.in	$68.00
85BYGH450C-03	NEMA 34	63kg.cm/874oz.in	$89.00

Part #:	Dim:	MicroStep:	Price:
XCW220	100 x 61 x 19mm	1(200), 1/2(400), 1/4(800), 1/8(1600)	$39.95
CW230	115 x 72 x 32mm	1(200), 1/2(400), 1/4(800), 1/8(1600), 1/16(3200), 1/32(6400), 1/64(12800)	$48.50
CW250	140 x 94 x 45mm	1(200), 1/2(400), 1/8(1600)	$54.90
CW860	147 x 97 x 30mm	1(200), 1/2(400), 1/4(800), 1/8(1600), 1/16(3200), 1/32(6400), 1/64(12800), 128(25600), 1/5(1000), 1/10(2000), 1/25(5000), 1/50(10000), 1/125(25000), 1/250(50000)	$96.00

Mini RF Transmitter, Receiver & Transceivers

Ideal for setting up short range wireless links for remote control or data acquisition!

- **Transmitter Module** w/Power Amplify
- Operating supply voltage: 5-12V
- Frequency: **418MHz**
- Frequency tolerance: ±300KHz
- Modulation: ASK/OOK
- Controlled by SAW device
- Antenna included

Only $9.95!

Item # STPA-418H-B

- **Receiver Module**
- Operating supply voltage: 5V
- Frequency: **418MHz**
- Sensitivity: -102dBm
- Band wide: 3.0MHz (-3dB)

Item # RXB4411S-418-RH

Only $15.95!

- **Programmable Transceiver**
- Operating supply voltage: 1.8-3.6V
- Frequency range: **300-928MHz**
- Data rate: 1.2-500kbps
- Output power: -30 to +10dBm
- Programmable via SPI
- Antenna included

Only $9.95!

Item # CC1100B

FLUKE TRMS Electronic Logging DMM w/ TrendCapture

The Fluke 287 True-rms Electronics Logging Multimeter with TrendCapture quickly documents design performance and graphically displays what happened. Its' unique logging and graphing capabilities mean you no longer need to download logged readings to a PC to detect a trend. This item is **Limited to Stock on Hand**!.

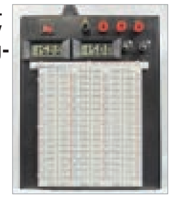
Item # FLUKE 287

Limited Offer!

Special Purchase Only $359.00!

PBB-272A Powered Breadboard w/ LCD Voltage Displays

Provides the user with a quick and efficient system for breadboarding electronic circuits. Comes with three regulated power supplies along with a deluxe, easy-to-use breadboard. **Two LCD's conveniently show the 0-15 positive and 0-15 negative outputs.**



- 5 Distribution Strips (500 tie points)
- 3 Terminal Strips (1890 tie points)
- 4 Binding Posts
  - One Ground
  - One 5VDC (1 AMP) Constant Power
  - One 0 to +15VDC (500mA) Variable Power
  - One 0 to -15VDC (500mA) Variable Power
  - 110VAC Input Power, ± 10%

Item # PBB-272A

Only $89.00!

Dual Output DC Bench Power Supplies

High stability digital read-out bench power supplies featuring constant voltage and current outputs. Short-circuit and current limiting protection is provided. SMT PC boards and a built-in cooling fan help ensure reliable performance and long life. All 3 Models have a **1A/5VDC Fixed Output** on the rear panel.

Item #:	Price 1-4	Price 5+
CSI3003X-5 0-30V/0-3A	$119.00	$112.00
CSI5003X5 0-50V/0-3A	$127.00	$119.00
CSI3005X5 0-30V/0-5A	$129.00	$122.00

- **Shielded Transceivers**
- DATA in RF out
- RF in DATA out
- Data rate: 1.2 to 10kbps
- Output power: +10dBm
- Sensitivity: -108dBm
- Operating supply voltage: 5V
- Frequency: **915MHz or 2.4GHz**
- Antenna included

A

Only $19.95 Each!

Item # EZ915 (A) 915MHz and Item # EZ2400 (B) 2.4GHz

B

Soldering Station w/Ceramic Element & Seperate Solder Stand

- Ceramic heating element for more accurate temp control
- Temp control knob in F(392° to 896°) & C(200° to 489°)
- 3-prong grounded power cord/static safe tip
- Seperate heavy duty iron stand
- Replaceable iron/easy disconnect
- Extra tips etc. shown at web site

Item # CSI-STATION1A

$39.95!

***Also* Available w/Digital Display & MicroProcessor Controller**

Item # CSI-STATION2A

SALE! $44.95

ESD Safe CPU Controlled SMD Hot Air Rework Station

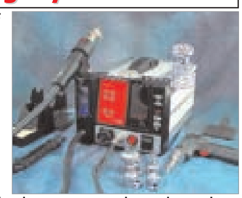
The heater and air control system are built-in and adjusted by the simple touch of the front keypad for precise settings. Temperature range is from 100°C to 480°C / 212°F to 896°F, and the entire unit will enter a temperature drop state after 15 minutes of non-use for safety and to eliminate excessive wear.

SALE! $129.00

Item# CSI825A++

Our Premium All in One Repairing System

- Combines the function of a Hot Air Gun, Soldering Iron and a Desoldering Gun.
- Microprocessor controlled ESD safe unit with all digital display
- Desoldering tool comes with zero crossing circuitry preventing electrical surges and equipped with air cylinder type strong suction vacuum pump.
- The 24V soldering iron is compatible with the compound tip design.
- Uses lead-free or standard solder.

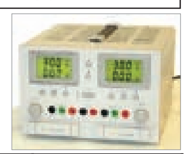


SALE! $199.00

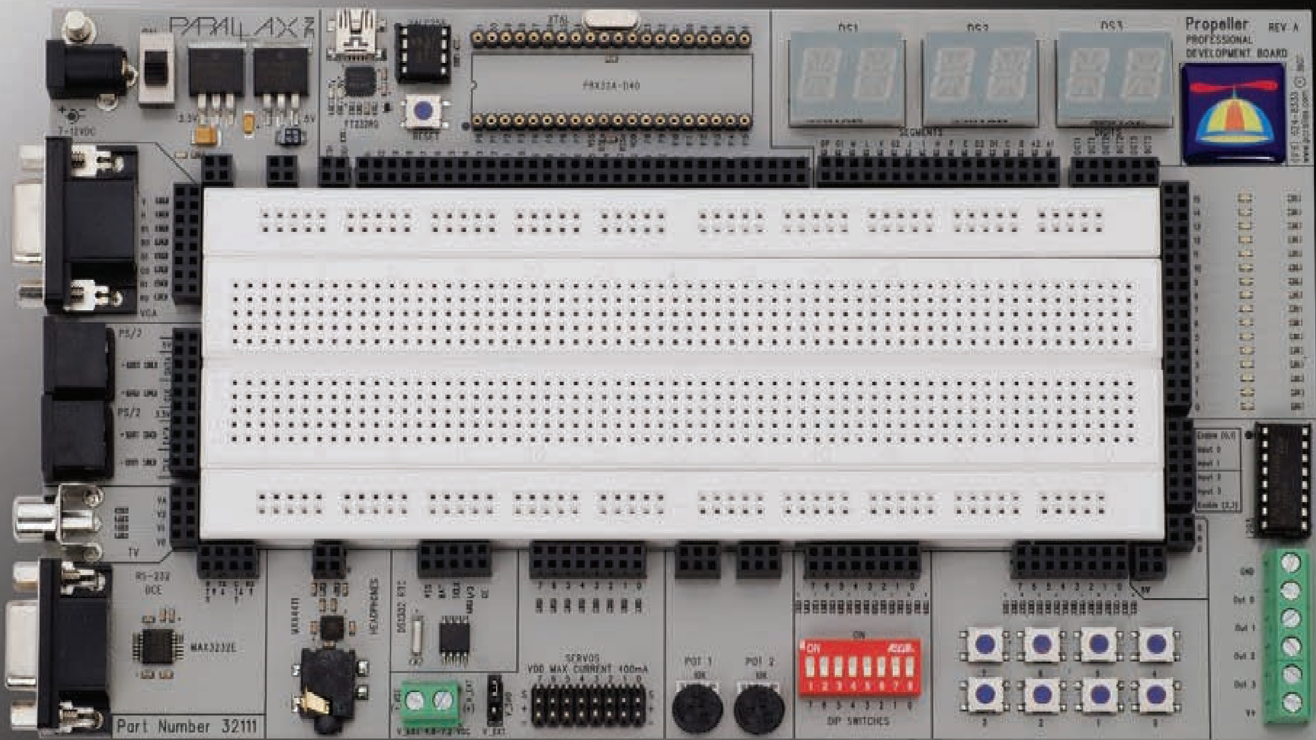
Item# CSI-9000

Triple Output DC Bench Power Supplies

- Output: 0-30VDC x 2 @ 3 or 5 Amps & 1fixed output @ 5VDC@3A
- Stepped Current: 30mA +/- 1mA



Item #:	Price 1-4	Price 5+
CSI3003X3 0-30Vx2@3A	$198.00	$193.00
CSI3005XIII 0-30Vx2@5A	$259.00	$244.00



new! PROPELLER PROFESSIONAL DEVELOPMENT BOARD (PPDB)

The Propeller Professional Development Board (PPDB) is a high-quality, fully-integrated development platform for the Propeller Chip. A wide variety of typical I/O (LEDs, buttons, etc.) devices and circuitry are built into the PPDB, providing the developer with an ideal platform for rapid Propeller Chip development. A 40-pin socket allows easy development and replacement using a through-hole version of the Propeller Chip (sold separately).

Features of the PPDB Include:

- 40-pin DIP socket for Propeller chip (#P8X32A-D40 sold separately)
- 32 K EEPROM (24LC256 included) in socket
- Socketed Crystal (5 MHz included)
- 6-digit, 16-segment LED Display (Blue, RHDP)
- 16 blue LEDs
- L293D quad push-pull driver
- 8 push-buttons
- 8 DIP switches
- 2 potentiometers (10 K)
- 8 servo headers with selectable internal/external voltage source
- DS1302 real-time clock with backup battery input (battery not included)
- Stereo headphone amplifier with 1/8" (3.5 mm) stereo phone jack
- RS-232 line driver (MAX3232E)
- RCA jack for TV/Composite video output
- 2 PS/2 connectors for mouse/keyboard interfacing
- VGA connector
- 2.1 mm barrel power jack and power switch
- 5 V and 3.3 V power supply connections
- Ground Terminal
- On-board USB Interface (mini-B, 5-pin)

Note: The USB cable (mini-B) and power supply are sold separately. We recommend the 12 V/1 A power supply (#750-00007) for the PPDB.

PARALLAX INC

www.parallax.com

Order the new **Propeller Professional Development Board** (#32111; $169.99) online at www.parallax.com or call our Sales Department toll-free 888-512-1024 (M-F, 7am-5pm, PDT).